Similarity Measures for Query Expansion in TopX

vorgelegt von
Caroline Gherbaoui
am 25.09.2008

angefertigt unter der Leitung von
Prof. Dr. Gerhard Weikum

betreut von
Dr. Ralf Schenkel

begutachtet von
Prof. Dr. Gerhard Weikum
PD Dr. Holger Bast
Erklärung:

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und alle verwendeten Quellen angegeben habe.

Saarbrücken, den 25.09.2008

Einverständniserklärung

Hiermit erkläre ich mich damit einverstanden, dass meine Arbeit in den Bestand der Bibliothek der Fachrichtung Informatik aufgenommen wird.

Saarbrücken, den 25.09.2008
Abstract

TopX is a top-k retrieval engine for text and XML data. Unlike some other engines, TopX includes an ontology. This ontology allows TopX to use techniques like word sense disambiguation and query expansion, to search for words similar to the original query terms. These techniques allow finding data items which would be ignored for the original source query, due to missing of words similar to the query terms. The similarity of words is given via the weights of the relations connecting words. The underlying ontology of TopX is the WordNet ontology, but in 2007 there was a further ontology integrated, the YAGO ontology.

This thesis has three main focuses:

- Import of a new version of the YAGO ontology.
- Similarity computation for YAGO relations.
- Adaptations of the TopX procedures for word sense disambiguation and query expansion to the differences between the WordNet ontology and the YAGO ontology.

We demonstrate the improvement of our approach for TopX, with center to the newly available YAGO ontology.
# Contents

Chapter 1: Introduction ................................................................. 1  
  1.1 Motivation .................................................................................. 3  
  Scalability: 3  
  Similarity Measures 3  
  1.2 Related Work ............................................................................... 3  
  1.3 Contributions .............................................................................. 5  
    1.3.1. New YAGO Version............................................................... 5  
    1.3.2. Similarity Measures .............................................................. 5  
    1.3.3. Word Sense Disambiguation.................................................. 6  
    1.3.4. Query Expansion................................................................. 6  
  1.4 Overview of this Thesis.............................................................. 6  

Chapter 2: Background ................................................................. 7  
  2.1 The WordNet Ontology .............................................................. 7  
  2.2 The YAGO Ontology .................................................................... 9  
    2.2.1. WordNet in YAGO ............................................................... 10  
    2.2.2. The Binding between YAGO and WordNet.......................... 12  
  2.3 The Ontology within TopX .......................................................... 13  
    2.3.1. The Word Sense Disambiguation.......................................... 14  
    2.3.2. The Query Expansion.......................................................... 17  
    2.3.3. WordNet within TopX ......................................................... 19  
    2.3.4. YAGO within TopX.............................................................. 20  

Chapter 3: Changes for the New YAGO Version ............................. 21  
  3.1 The Dijkstra Algorithm ............................................................... 21  
  3.2 The Configuration File ............................................................... 24  
    3.2.1. Connection Data for the Ontology Database ....................... 24  
    3.2.2. Types of Relations .............................................................. 24  
    3.2.3. Types of Relations for the Query Expansion ....................... 25  
    3.2.4. Switching between YAGO and WordNet............................ 25  
  3.3 Extensions for the YAGO Ontology ............................................ 26  
    3.3.1. Import of Value Relations.................................................. 26  
    3.3.2. Two Kinds of Word Reducing ............................................ 27  
    3.3.3. Import of New Words ........................................................ 27  
  3.4 Statistics of the YAGO Ontology ................................................. 28
## Chapter 4: Similarity Measures

4.1 Dice Similarity .................................................................................................................. 32
  4.1.1 Default Definition ........................................................................................................ 32
  4.1.2 Definition in TopX ........................................................................................................ 33
4.2 NAGA Similarity ................................................................................................................. 34
  4.2.1 Default Definition ........................................................................................................ 34
  4.2.2 Definition in TopX ........................................................................................................ 36
4.3 Best WordNet Similarity .................................................................................................... 38
  4.3.1 Default Definition ........................................................................................................ 38
    The Transfer Function of the Path Length ........................................................................ 40
    The Transfer Function of the Abstractness / Concept Depth ........................................... 40
    Final Similarity Measure .................................................................................................. 41
  4.3.2 Definition in TopX ........................................................................................................ 41
4.4 Worked Samples .................................................................................................................. 46
  Dice Similarity ....................................................................................................................... 46
  NAGA Similarity .................................................................................................................. 47
  Best WordNet Similarity ...................................................................................................... 48
4.5 Exceptions in the Similarity Computation ......................................................................... 48
  4.5.1 The Binding between YAGO and WordNet ................................................................. 49
  4.5.2 Value Relations ........................................................................................................... 50
  4.5.3 Special Signs ............................................................................................................... 50
4.6 Efficient Computation ........................................................................................................ 51
4.7 Overview ............................................................................................................................ 52

## Chapter 5: Evaluation of the Computed Similarity Values

5.1 Best WordNet measure ..................................................................................................... 54
5.2 NAGA measure .................................................................................................................. 54
  The Influence of the Confidence .......................................................................................... 56

## Chapter 6: Word Sense Disambiguation at the YAGO ontology

6.1 The Problem ..................................................................................................................... 59
6.2 Methods of Resolution ...................................................................................................... 60
  6.2.1 WordNet Senses ......................................................................................................... 60
  6.2.2 Wikipedia Senses ....................................................................................................... 60

## Chapter 7: Query Expansion for the YAGO Ontology

7.1 The Word Expansion .......................................................................................................... 68
7.2 NAGA Similarity ............................................................................................................... 68
7.3 Worked Samples ............................................................................................................... 71
  Dice Similarity Measure on the WordNet ontology ............................................................. 71
  Dice Similarity Measure on the YAGO ontology ................................................................. 71
  NAGA Similarity Measure on the YAGO ontology .............................................................. 72
### Chapter 8: Conclusions

8.1 Open Issues & Future Work ................................................................. 74  
8.1.1. Word Sense Disambiguation...................................................... 74  
8.1.2. Query Expansion ................................................................. 75  
8.1.3. Value Relations ................................................................. 75  
8.1.4. Relation Similarities ............................................................ 76  
8.1.5. Evaluation of the Query Expansion ........................................... 76  

### Appendix A

A.1. TREC Terabyte Queries ............................................................. 79
Chapter 1

Introduction

Top-k query processing is an "important building block for ranked retrieval" [25] and is also used for many other kinds of information discovery like multimedia similarity search or preference queries over product catalogs, as mentioned in [26]. Top-k queries compute the k most relevant results to a partial-match query, based on similarity scores.

Normally, in case a relevant document does not contain the terms out of the query, this document would not be retrieved. To this end there exists the query expansion [27], [28], [29]. It reduces this problem by expanding the query terms with words or phrases which are semantically or statistically related to the original query terms.

However, some of the original query terms could have multiple meanings (i.e., polysemous terms). Before searching similar terms, the correct meaning of these terms is needed; otherwise, the query expansion would lead to wrongly related terms. This process is called word sense disambiguation [30]. There all possible meanings of a term are taken into account and the correct one of them is chosen. Only the correct meaning is passed to the query expansion.

These two techniques can enhance the quality of the results of a top-k query engine. However, they require a knowledge base, the ontology, which contains an amount of terms together with their meanings and their semantic or statistic relations.

An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. Moreover, there are individuals, classes, and relations. Individuals are instances of objects, whereas classes are types of objects. The relations connect classes and individuals to each other.

We consider the following example; a user searches for data items to the source query "java, coffee":


Figure 1-1 shows that, in the default case, the top-k engine searches only for data items including terms of the query "java, coffee". If the user chooses expansion, the query expansion adds the similar words "espresso, drink, …" to the source query terms using the ontology. In this case the search engine considers the source terms "java, coffee" plus the related terms "espresso, drink, …".

The amount of terms which will be added during the query expansion depends on the similarity between the respective terms in the ontology. These similarities are the weights of the relations. That means, if one term is very similar to a term, the respective relation has a higher similarity than a relation to a less similar term.

In this thesis we consider TopX [1], [2], which is a top-k retrieval engine for efficient, ranked retrieval of text and XML data. Moreover, TopX uses the previously mentioned techniques, the word sense disambiguation and the query expansion; thus, it also contains an ontology. In this underlying ontology, words are related to each other in terms of synonyms, subclasses, superclasses, and many more. In the past, the WordNet ontology [3], [4] was the sole ontology for TopX (see also Section 2.1). WordNet is a large lexicon for the English language, including relations between its synsets, as for example $a$ is a synonym of $b$, or $a$ is a kind of $b$. Since 2007, a further ontology exists, namely the YAGO ontology [5], [7] (see also Section 2.2). Its words and relations are derived from Wikipedia and a part of WordNet.
1.1 Motivation

**Scalability:**

At the import of the YAGO ontology in 2007 [18], the required database tables were created and the access to these database tables was implemented in TopX. The techniques for word sense disambiguation and query expansion, were left unchanged as far as possible. However, with the consideration that the YAGO ontology is much bigger than the WordNet ontology and that the YAGO ontology also has other kinds of relations (e.g., bornin), we study the implementations of these techniques in TopX to discover whether they are still applicable for the YAGO ontology.

**Similarity Measures**

Another issue are the similarity values for the query expansion (i.e., the relation similarities). As already mentioned, the number of terms which will be expanded during the query expansion depends on the similarity values between the terms in the ontology. In this thesis we consider the currently used similarity measure in TopX, the Dice similarity, in relation to the YAGO ontology. Moreover, we search for further similarity measures, mentioned in several papers for information retrieval, and try to adapt them to the needs of TopX and YAGO. Afterwards we compare the computed similarity values for all measures on the YAGO ontology with the computed similarity values for the Dice similarity measure on the WordNet ontology. In this manner, we will show the practicability of each measure with center to the YAGO ontology.

YAGO offers a new version since March 2008. Since we want the similarity computation bases on the newest version of YAGO; we first deal with the import of the new version of YAGO.

1.2 Related Work

The problem of finding the best similarity measure for the semantic similarity, or, more generally, relatedness, between two lexically expressed concepts or between words, pervades much of computational linguistics.

As mentioned in many papers like in [14], similarity measures are used in applications like *word sense disambiguation, determining discourse structure, text summarization and annotation, information extraction and retrieval,*
automatic indexing, lexical selection, and automatic correction of word errors in text. In this thesis we focus on the similarity between concepts and words.

Most of the measures are tied to a particular application or assume a particular domain model. As an example see Alexander Budanitsky and Greame Hirst [14]. They evaluate different measures for the WordNet domain. The fact that these measures are constructed for the WordNet domain qualifies them to be adequate candidates for the similarity computation on the YAGO relations.

The similarity measures for WordNet follow different approaches. One known approach is the corpus-based approach. There, word similarities are often derived from their co-occurrences in a corpus [31]. The Dice similarity measure [15], the already used one in TopX (see Section 4.1), belongs also to this kind. Other approaches use information derived from the lexical taxonomy [32], [33]. One measure, which uses corpus and ontology knowledge and is also listed in many other papers, is the measure of Resnik [8]. He defines the similarity of two concepts by the information content of their most specific common subsumer. The measure of Jiang-Conrath [21] obtained best results in the paper of Alexander Budanitsky and Greame Hirst; however it is not the best measure belong all available WordNet measures. It is a combination of an edge-based approach, by counting edges, and a node-based approach, by the information content. The measure, which achieved best results among all WordNet measures until 2007, is the measure from Yuhua Li at al. [19] (see Section 4.3). It combines ontology knowledge only.

A further interesting and significant measure for this thesis, is the NAGA measure [17] from Gjergji Kasneci et al. This measure is special for the YAGO ontology. It combines the confidence, informativeness and compactness of a relation.

Another interesting definition of the similarity is the thesis of Dekang Lin [13]; he defines the similarity measure that way that it is independent on the application and the knowledge information. He states, that the similarity between two concepts bases on the ratio between the amount of information needed to state the commonality between those concepts and the information needed to fully describe what the concepts are.

That means, there exist a lot of different similarity measures; however, the challenge is to adapt an available measure to the own application or the own domain model.
1.3 Contributions

1.3.1. New YAGO Version

First of all, we provide the newest version of YAGO within TopX. The import of this version entails some changes in the implementation of some procedures of TopX, as for example the Dijkstra algorithm (see Section 3.1). A further enhancement is the extension of the configuration file (see Section 3.2). To this manner we make TopX more flexible, with regard to the second available ontology, the YAGO ontology.

Moreover, we provide several procedures for the later import of further YAGO versions or parts of YAGO (see Section 3.3).

To show the differences between both YAGO versions (i.e., the version from December 2007 and the version from March 2008), we list some statistics of both versions (see Section 3.4).

1.3.2. Similarity Measures

The Dice similarity measure is the currently used one in TopX. In this thesis we apply this measure at the computation of the similarity values on the YAGO relations. Moreover, we introduce the following two measures for the similarity computation on the YAGO ontology:

- The NAGA similarity measure [17].
- The Best WordNet similarity measure (until 2007) [19].

Prior to the implementations we adopt their definitions to the requirements of TopX and YAGO (see Section 4.2 and 4.3). Thereby we establish some exceptions on the similarity computation due to the structure of the YAGO ontology (see Section 4.5). Moreover, we present a special implementation of the similarity computation for an efficient and consequently fast computation of the similarity values (see Section 4.6), due to the huge amount of relations in YAGO.

Afterwards, we evaluate the computed similarity values and compare them to each other, to get evidence whether they can be applied within TopX (see Chapter 5).
1.3.3. Word Sense Disambiguation

The word sense disambiguation on the YAGO ontology, as defined for the WordNet ontology, is not applicable, due to the size of the YAGO ontology. In this thesis we analyze and stage the problem appearing thereby (see Section 6.1).

Furthermore, we offer some approaches for the resolution and stabilize our efforts with several tests (see Section 6.2). Our best approach achieves results equal to best results of the word sense disambiguation on the WordNet ontology.

1.3.4. Query Expansion

In this thesis we analyze the query expansion through the YAGO ontology. Since the words of YAGO contain special signs, as for example Arabic letters, we adapt the query expansion in this context (see Section 7.1).

Moreover, we show the changes on the query expansion for the NAGA measure, due to the structure of this measure (see Section 7.2).

1.4 Overview of this Thesis

The rest of the thesis follows the following structure. Chapter 2 describes the WordNet ontology and the YAGO ontology. It also depicts the integration of both ontologies in TopX. Moreover, it describes the procedures, word sense disambiguation and query expansion, as they work on the WordNet ontology. Chapter 3 covers the import of the new YAGO version. Including changes on existing code, upgrading of the configuration file and further implemented procedures for the aid of later imports of the YAGO ontology. In this chapter we confront the data statistics of both versions of YAGO, December 2007 and March 2008. Chapter 4 introduces the three consulting similarity measures and shows their definitions for TopX. One is the already existing measure in TopX, the Dice similarity. The both other measures, NAGA and the Best WordNet measure, are obtained from other papers. Chapter 5 evaluates the computed similarity values, based on the three measures. There the similarity values for each measure are compared to each other. Chapter 6 investigates with the problem of the word sense disambiguation on the YAGO ontology and the corresponding methods of resolution. Chapter 7 occupies with the query expansion on the YAGO ontology. Chapter 8 concludes this thesis.
Chapter 2

Background

2.1 The WordNet Ontology

WordNet [3], [4] is a large semantic lexicon for the English language. It contains nouns, verbs, adjectives, and adverbs which are grouped into sets of cognitive synonyms. These synonyms are called words, where such a word can be a single word or a number of words, as for example "course of action". A set of words which are considered semantically equivalent for the purposes of information retrieval (i.e., synonyms) is called a synset. Moreover, words with multiple meanings belong to multiple synsets, which are identified by IDs. The meaning of a synset is assigned by an attached gloss which is a natural linguistic description.

In other words, a synset is a semantic meaning and the words of this synset are words which can be used for this meaning.

We show an example of a synset in Figure 2-1, where "00007846" is the identification of the synset and "a human being" is the attached gloss. The strings inside the circle are the words for this synset.

The synsets are related by means of conceptual-semantic and lexical relations and are completely connected through the IS-A hierarchy. Conceptually, the relation hypernym (i.e., the relation between a sub-concept and a super-
concept) in WordNet spans a directed acyclic graph (DAG) with 9 root nodes. Moreover, the relations between two given synsets are valid between any combinations of their respective words, as follows:

If \( r \) is a relation between synset \( s_1 \) and synset \( s_2 \), then it can be generalized to a relation \( r' \) for words with

\[
r'(x, y) \iff \exists_{s_1, s_2}, \text{ such that } x \in s_1, y \in s_2, r(s_1, s_2)
\]

(2.1)

Not only the type of a relation (e.g., hypernym) is called relation, but also the triple consisting of a synset \( x \), a type of relation \( r \), and a synset \( y \), is called relation. Here is an example for such a triple:

1000001 is a hypernym (i.e., the relation between a sub-concept and a super-concept) of 1000002, where hypernym is the relation, and "1000001" and "1000002" are two distinct synsets. All words in the synset "1000001" are related, through the relation type hypernym, to all words in the synset "1000002".

In Figure 2-2 the circles illustrate synsets and the squares within the circles illustrate the words of the synsets. The solid arrow depicts the relation between the synsets, in this case hypernym. The dashed arrows denote the generalization of the relation hypernym to the words of the synsets.

In version 3.0, the WordNet ontology contains 155,287 words for 117,659 synsets and 206,941 pairs of synsets (i.e., instances of relations).
2.2 The YAGO Ontology

The YAGO ontology [5], [6], [7] is a large ontology derived from Wikipedia [23] and WordNet. All objects (e.g., cities, people, even URLs) are represented as entities in the YAGO model. Numbers, dates, strings, words and other literals are represented as entities as well and we call this kind of entities values. Two entities can stand in a relation. The triple consisting of an entity, a relation and an entity is called a fact, where every fact has its own ID, the fact identifier. More than 1.7 million entities and over 15 million facts are currently contained in the YAGO ontology. Moreover, objects are comparable with the synsets of the WordNet ontology. YAGO is automatically derived and has been extracted from the category system and the info boxes of Wikipedia. Also taxonomic relations from WordNet are included in the YAGO ontology, which is available as plain text files. The facts from YAGO are located in the folder "facts", where the subfolders of the folder "facts" denote the actually valid relations in YAGO. Currently 96 different relations exist in YAGO\(^1\). Unlike in WordNet a fact can be an entity as well. That means facts can illustrate almost everything in YAGO.

Here are some examples of facts \(r = (x, t, y)\) (also called relations), where \(x\) and \(y\) are entities and \(t\) is the relation type:

- **subclassof** is a relation between two entities, where both entities are objects.
  For example, \(((\text{US State}), \text{subclassof}, (\text{wordnet_state_108125703}))\) (see Figure 2-3).
  We call this kind of relations, object relations.

- **means** is a relation between two entities, where one entity is a word and the other entity is an object.
  For example, \((\text{"US State"}, \text{means}, (\text{US State}))\) (see Figure 2-3).
  We call this kind of relations, value relations\(^2\). The words of the objects are assigned by the value relations *familynameof*, *givennameof*, *is-called*, *isnativenameof* and means.

- **since** is a relation between two entities, where one entity is a fact (represented by a fact identifier), and the other entity is a date.
  For example, \(((100000), \text{since}, \text{"22.09.2002"})\); 100000 is the fact identifier of the fact \(((\text{Angela Merkel}), \text{type}, (\text{Chancellor}))\) (see Figure 2-4).
  We call this kind of relations, fact relations\(^3\).

---

\(^1\) Not all of them are relevant for TopX
\(^2\) Currently, there exist 36 different value relations.
\(^3\) This kind of relations is not relevant for TopX and therefore we will not go into detail for this kind in this thesis.
Chapter 2 Background

Conceptually, the relations \textit{subclassof} and \textit{type} (i.e., the relations between a sub-concept and a super-concept) in YAGO span - like the relation \textit{hypernym} in WordNet - a directed acyclic graph (DAG).

2.2.1. WordNet in YAGO

The YAGO ontology includes a part of the WordNet ontology. This part consists of an amount of nouns and a number of relations of WordNet; the latter are shown in Table 2-1. Relation names in YAGO differ from the relation names in WordNet. For this purpose we list the relation names in WordNet on the left side and the corresponding relation names in YAGO on the right side of the Table 2-1.

<table>
<thead>
<tr>
<th>Relation names in WordNet</th>
<th>Relation names in YAGO</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypernym</td>
<td>hyponym</td>
</tr>
<tr>
<td>member-meronym</td>
<td>member-holonym</td>
</tr>
</tbody>
</table>
There is a further important difference between YAGO and WordNet. As observable in Table 2-1; two relations in WordNet correspond to one relation in YAGO.

That means, in WordNet we have for the forward- and backward direction each one relation (and both are opposed to each other). See Figure 2-5, where the relation **hypernym** is the forward relation and the relation **hyponym** is the opposite and thus the backward relation.

![Figure 2-5](image)

YAGO relations have to be read *from the left to the right*. The name for the backward relation of **subclassof** would be "superclassof". That means, if we have the fact \(A, \text{subclassof}, B\), then we know \(A\) is a subclassof \(B\) and \(B\) is a superclassof \(A\). In other words, we get the name of the backward relations by the opposite of the YAGO relation names. These backward relations are not included in the YAGO ontology, since each backward relation can be obtained by visiting the respective relation backwards. In Figure 2-6 is shown, that the YAGO relation **subclassof** combines both WordNet relations **hypernym** and **hyponym**, where the backward relation is left implicit.

![Figure 2-6](image)

**Notice:** In Table 2-1, relations for the version of March 2008 are shown. Since the extension of the YAGO ontology is not finished yet, this list might probably change in the future.
2.2.2. The Binding between YAGO and WordNet

As Figure 2-7 depicts, the YAGO ontology consists of two levels, the WordNet level and the Wikipedia level. The red arrows within the upper area illustrate the relations in the YAGO ontology which were extracted from WordNet. These relations bind only WordNet entities (see also Section 2.2.1), with the exception of the relation subclassof. The blue arrows within the lower area illustrate the relations from the YAGO ontology which bind only typical YAGO entities, so-called Wikipedia individuals. The light green arrows from the shadowed area to the upper area illustrate the facts of the relation subclassof - also illustrated by the dashed line labeled with "subclassof" - which bind the Wikipedia level with the WordNet level. The bold green arrows from the lower to the shadowed area illustrate the semantic relation type - also illustrated by the dashed line labeled with "type". On the right side within the figure is observable that the relation type also binds the Wikipedia level with the WordNet level, directly.

As already mentioned above the relation subclassof operates not only as equivalent to the relations hypernym and hyponym (see also Section 2.2.1), but also as binding between the Wikipedia level and the WordNet level.

The Wikipedia level itself is also divided into two layers; the shadowed area is a middle layer and illustrates the Wikipedia categories. However, currently this layer is not of interest for TopX and will always be skipped in TopX. The lower layer is the "real" Wikipedia level which consists of Wikipedia individu-
als. Figure 2-8 shows an example of the relationship for "Albert Einstein" within the YAGO ontology.

2.3 The Ontology within TopX

With the intention to enhance the ranked results for a given query, TopX includes an ontology whereby the given query can be expanded with additional query terms which are semantically similar to the original query terms.

In TopX, WordNet was the sole ontology for a long time. Within the thesis Integration of the YAGO Ontology into the TopX Search Engine [18] a second ontology, the YAGO ontology, was integrated into TopX. Classes like the synsets of WordNet are also called concepts. Moreover, for simplification we call a Wikipedia entity (which can be a Wikipedia category, a Wikipedia individual or an entity in the WordNet level) in this thesis, concept. In order to decide whether a related concept is more important than another related concept, the edges of the ontology graph are weighted with a similarity value for the respective concepts; there a related concept over a higher weighted edge is more important than a related concept over a smaller weighted edge.

Since query terms could yield to different concepts in the ontology, due to polysemous terms, first of all a disambiguation over all possible concepts has to be done, the so-called word sense disambiguation. Having the correct con-
cepts disambiguated, semantically similar concepts for these concepts have to be searched for. The words of the found similar concepts can be added to the original query terms. This procedure is called *query expansion*.

In this section, first of all, we explain the word sense disambiguation and the query expansion, as defined for the WordNet ontology, since these are the major functions which are engaged in the finding additional query terms. Finally, we show how both ontologies, the WordNet ontology and YAGO ontology, are included in TopX.

### 2.3.1. The Word Sense Disambiguation

As already explained in [2], we have several terms in a query. We start with considering a single term out of the query, for example the term "java". This term leads to the following different word senses in the YAGO ontology:

1) \{java, coffee\} - a beverage consisting of an infusion of ground coffee beans; "he ordered a cup of coffee".
2) \{java, java dance\} - The Java is a dance developed in France in the early part of the 20th century. …
3) \{java, java technology, java™, java applets, sun java, …\} - Java refers to a number of computer software products and specifications from Sun Microsystems (the Java™ technology) that together provide a system for developing and deploying cross-platform applications. …
4) \{java, java programming, java (software), …\} - … Java is an object-oriented programming language developed by Sun Microsystems in the early 1990s. …

And some more senses.

The first sense is a WordNet concept, and the remaining are Wikipedia concepts. Each of these concepts is connected to different concepts in the ontology graph. Figure 2-9 and Figure 2-10 depicts that the environment of each concept can have variable sizes.
A reliable disambiguation on these possible senses has to be done to find the right sense of the word "java". This approach - finding the right one among the possible senses of a word - is called word sense disambiguation (WSD).

There exist several different approaches. As mentioned in [2], TopX uses a so-called *unsupervised* approach (i.e., fully automatic [2]), the *Independent Mapping*.

Not only single terms out of the given query were used as source for the disambiguation, but also n-grams out of the query terms. To explain how TopX achieves the largest possible subsequence of n-grams out of a given query, we consider the word sequence \( w_i, \ldots, w_m \) of the query. The first word \( w_i \) in the sequence is at position \( i = 1 \). A small look ahead distance \( m' \) of at most 5 words is used. TopX uses a simple window parsing technique to determine the largest subsequence of words, which can be matched with a phrase contained in concepts: Every concept which contains such a phrase (i.e., the subsequence of words out of the query) will be added to the set of possible word senses, \( S_{p_i}, \ldots, S_{p_s} \) for this phrase. If the current sequence, \( w_i, \ldots, w_{i+m'} \), has been successfully matched, \( i \) will be incremented by \( m'\) and the mapping procedure will be continued on the suffix of the sequence; if the current sequence could not be matched onto any phrase denoted by a concept, \( m' \) will be decremented by 1 and the lookup will be tried again until \( m' = 1 \). After performing that subroutine, \( i \) will again incremented by 1 until \( i = m \).

Figure 2-11 shows the windows parsing technique on the example query "U.S. against International Criminal Court". There "U.S. against International" is the first subsequence for the subroutine with the look ahead distance \( m' = 3 \).
Chapter 2 Background

Figure 2-11

To discover the right sense of a word or a n-gram t, the independent Mapping regards the local context \( con(t) \) of t, which is the set of the query terms exclusive of the word or the n-gram, respectively. For comparison, the local context of each candidate word sense \( S \) will be considered. These local contexts include the words of the respective sense and all immediate hyponyms and hypernyms up to depth two. Since all of these hyponyms and hypernyms are also concepts and have also a description, their words and their respective descriptions will be added, to get the complete local context \( con(S) \) of a sense \( S \). As an example, the context of the first sense of the word "java" is the bag of words \{java, coffee, drink, ..., caffeine, mocha coffee, ...\} (the bold words are the own words and the other words are the words of the hypernyms and hyponyms) plus the description of the included hyponyms and hypernyms of the first sense (see Figure 2-9), whereas the second sense gets the bag of words \{java, java technology, ..., platform, ...\} plus the description of the included hyponyms and hypernyms of the second sense (see Figure 2-10).

As final step the independent Mapping compares the context \( con(t) \) of the considered term or n-gram t with each context \( con(S_1), ..., con(S_n) \) of the candidate senses \( S_1, ..., S_n \). The comparison is implemented by the standard IR measure Cosine similarity between \( con(t) \) and \( con(S_j) \), where \( S_j \in \{S_1, ..., S_n\} \).
If the similarity value is zero for all candidate senses, the candidate with the most frequently word is taken.

### 2.3.2. The Query Expansion

After word sense disambiguation is done, we have the correct source concepts which we use to search for additional query terms. If semantically similar concepts are found, their words will be added to the original query terms; this process is called *query expansion*.

To find semantically similar concepts, TopX does, out of the disambiguated concepts, a retrieval on the ontology graph for concepts whose similarity value is greater or equal a defined value $\delta$.

To this end, we have to regard the similarity of paths, since all concepts found are related to the source concept through a relation path. In TopX, the similarity value of paths is based on the maximum likelihood of independent successive relations.

Hence, if we have the path $\langle r_1, \ldots, r_m \rangle$ of length $m$, which connects the source concept $s$ with the target concept $t$, the final path similarity value $\text{sim}(\langle r_1, \ldots, r_m \rangle)$, and so the similarity value of the source concept and the target concept $\text{sim}(s, t)$, is defined as follows:

$$\text{sim}(\langle r_1, \ldots, r_m \rangle) = \text{sim}(s, t) = \prod_{i=1}^{m} \text{sim}(r_i)$$  (2.2)

The similarity values $\text{sim}(r_i)$ of the single relations $r_i$ out of the path $\langle r_1, \ldots, r_m \rangle$ are the precomputed weights of the edges (i.e., the similarity values of the relations).

Moreover, we have to note that the expansion considers only one type of relation. That means, if the query expansion should be done for several relation types (e.g., for *subclassof* and *ismemberof*), the expansion will be done for each relation type, separately.

We have the following procedure for the query expansion:

1) For each concept $d$ out of the source concepts $s_1, \ldots, s_n$ do the following:
2) As long as outgoing edges $e_{d,j}$ of concept $d$ exist:
   3) Compute the similarity value $\text{sim}(d, t_j)$ of target concept $t_j$ of edge $e_{d,j}$, with $\text{sim}(d, t_j) = \text{sim}(d) \cdot \text{sim}(e_{d,j})$. 

17
Chapter 2 Background

If concept $d$ is a source concept, then $\text{sim}(d) = 1$ and 
$\text{sim}(d, t_j) = \text{sim}(e_{d,j})$.

4) If $\text{sim}(d, t_j) \geq \delta$, add the words of $t_j$ to the query terms, and go to step 2) for concept $t_j$.

If $\text{sim}(d, t_j) < \delta$, in this case the query expansion is stopped for the subgraph spanned by the edge $e_{d,j}$.

Since the similarity value of each edge is in the range $[0, 1]$ (i.e., 0 for no similarity, and 1 for synonymous concepts), every step of the query expansion returns a similarity value equal or smaller to the previous similarity value. Hence, this approach provides exactly those concepts whose similarity values are greater or equal $\delta$ with minimal graph retrieval.

![Diagram](image)

Figure 2-12

As an example, the expansion for the source concept $A$ is shown in Figure 2-12. The words of the concept $C$ will be added to the query terms and the expansion will search for further similar concepts in the subgraph spanned by the outgoing edges of concept $C$. The words of the concept $D$ will not be added to the query terms, since the similarity value of the concept $D$ with 0.25 is smaller than $\delta = 0.5$, and the subgraph spanned by the outgoing edges of $D$ will not be considered.
2.3.3. WordNet within TopX

TopX keeps the WordNet ontology within a database. The table SYNSETS contains all synsets with their words.

The Table 2-2 shows an entry of the SYNSETS table.

<table>
<thead>
<tr>
<th>concept</th>
<th>conceptstem</th>
<th>conceptfreq</th>
<th>csynsetid</th>
<th>pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>intravenous feeding</td>
<td>intraven feed</td>
<td>504</td>
<td>20806188</td>
<td>n</td>
</tr>
</tbody>
</table>

The descriptions of the synsets are contained in the table DESCRIPTIONS. There the synset IDs with the respective gloss are saved.

The Table 2-3 shows an entry in the table DESCRIPTIONS.

<table>
<thead>
<tr>
<th>csynsetid</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20727297</td>
<td>a military training exercise</td>
</tr>
</tbody>
</table>

Further tables, named by the relation types, contain the synset pairs for each relation type (i.e., the edges of the ontology graph). To this end, the source synset and the target synset with their respective synset IDs and their respective words are saved. Moreover, the similarity values for each synset pair (i.e., edge weights) are also contained in these tables.

The Table 2-4 shows an entry in the table HYPERNYM.

<table>
<thead>
<tr>
<th>concept</th>
<th>conceptstem</th>
<th>conceptfreq</th>
<th>csynset-id</th>
<th>target</th>
<th>targetstem</th>
<th>targetfreq</th>
<th>tsynset-id</th>
<th>sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>intravenous injection</td>
<td>intraven inject</td>
<td>1164</td>
<td>20246638</td>
<td>fix</td>
<td>fix</td>
<td>49422</td>
<td>20246748</td>
<td>0.0014</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>287612</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>866610</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>289</td>
</tr>
</tbody>
</table>
2.3.4. *YAGO within TopX*

Like the WordNet ontology, TopX keeps the YAGO ontology within a database. The descriptions of the entities in the YAGO ontology are contained in the table `CONCEPTS`; this table is the counterpart to the table `DESCRIPTIONS` in the WordNet database. For the entities there were two tables introduced, the table `ENTITYIDS`, which contains the mapping between the YAGO entity names and the TopX IDs for the entities, and the table `SYONYMS`, which corresponds to the table `SYNSETS` in the WordNet database.

The storage of the YAGO facts is as follows: instead of several tables for several relation types, as in the WordNet database, there only one table `RELATIONS` exists. It contains the relation ID of the corresponding relation type, the source concept ID and the target concept ID. Moreover, the only similarity value of the WordNet edges is replaced by two similarity values, a forward similarity value (i.e., visiting the edge from left to right) and a backward value (i.e., visiting the edge from right to left) (see also Section 2.2).

The Table 2-5 shows an entry in the `RELATIONS` table.

<table>
<thead>
<tr>
<th>relationid</th>
<th>sourceconcept-id</th>
<th>targetconcept-id</th>
<th>forward-sim</th>
<th>backwardsim</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1000181541</td>
<td>1002047401</td>
<td>0.021223708987236</td>
<td>0.021223708987236</td>
</tr>
</tbody>
</table>

For the purpose of TopX only object relations are needed, thus the value relations and the fact relations were ignored during the importation. Moreover, due to the unimportant middle layer in the Wikipedia level (see also Section 2.2.2), all procedures walking through the ontology were adapted by ignoring this layer. Finally, a further table `RELATIONNAMES` were introduced to save the mapping between the relation IDs and their corresponding relation names.
Chapter 3

Changes for the New YAGO Version

In order to have the latest version for the similarity computation we started importing the newest version of YAGO from March 2008. This version differs from the version of December 2007 in two aspects; by the names of the relation types and by the number of entities, facts and words. In this section we will explain the changes and extensions we have done for TopX due to the new version.

One problem was the increase of data. Some procedures, which were implemented for the much smaller WordNet ontology, took too much time or exceeded the memory because of the size of the YAGO ontology. One of these procedures was the Dijkstra algorithm.

A further topic was the availability of the YAGO ontology, as second ontology. Some properties and ontology information were hard coded in TopX, since only one ontology existed at this time. For this purpose we exported some information from hard coded to entries in the configuration file.

3.1 The Dijkstra Algorithm

The Dijkstra algorithm computes the shortest distance between a source node $s$ and any other node $n$ in an edge weighted graph $g$, also called shortest path. In TopX, the Dijkstra algorithm can be used to compute the best concept candidates for the word sense disambiguation, where every edge is weighted with 1. Later we will show that it is also needed for the depth computation of concepts (see also Section 4.3).

The shortest paths computation in the Dijkstra algorithm works as follows:

Initial every node $n$ will be added to the priority queue with a very large depth $max$. The computation starts with the source node $s$, assigns it the depth 0 in the priority queue and marks the source node $s$ as visited. The following steps go recursively as long as nodes exist in the priority queue:
Chapter 3 Changes for the New YAGO Version

1) Take and remove node $m$ from the priority queue $k$ with minimum depth. If node $m$ was not yet visited continue with step 2) else go to step 1).
2) Mark node $m$ as visited.
3) Search all outgoing edges $e_1,...,e_i$ of $m$ and take their destination nodes $d_1,...,d_i$.
4) Adapt the node depths of the nodes $d_1,...,d_i$ to depth $\text{depth}(m)+1$.
5) Adjust depths of the nodes $d_1,...,d_i$ in the priority queue $k$.
6) Return to step 1).

The old TopX loaded the underlying graph for the Dijkstra algorithm as follows:

- It uses two hash maps for the mapping from the node ID to the concept ID and vice versa.
- For every node $n$ each outgoing and incoming edges were saved within a hash map. This resulted in doubled edge costs, since every edge was saved for the source node and the destination node.

The priority queue in TopX was implemented with the sorted set from java. There were several problems for the YAGO ontology:

- The YAGO ontology is very large; around 6 million relations and around 2 million nodes. Therefore it takes too much memory and too much time to compute the graph.
- Since the priority queue was implemented via a sorted set, every insert and delete operation on this priority queue caused a re-sort of the set, with time $O(\log(n))$ for each. Since the YAGO ontology has many concepts (which correspond to graph nodes), these operations were considerably slow.

We had to search a graph representation which takes less memory and an alternative for the priority queue which takes smaller expenditure of time for every insert and delete operation.

The new graph representation is realized as follows:

- We take one linked list for each node where the outgoing edges are saved and insert these lists in a second linked list where the list indices denote the node IDs. In other words, on index $n$ we have the outgoing edges of node $n$.
- One outgoing edge is realized as tuple $(a,b)$, where $a$ is the destination node ID and $b$ is the edge type (i.e., name of the relation type).
3.1. The Dijkstra Algorithm

Figure 3-1 depicts the new graph representation, which is an adjacency list construction, where the horizontal grey bar illustrates the node list and the vertical yellow bars illustrate the respective edge lists.

Instead of the priority queue we take a further linked list $l$ and add the source node on it. As with the priority queue the following steps go recursively as long as nodes exist in the list $l$:

1) Take and remove the first node $m$ from the list $l$.
   If node $m$ was not yet visited continue with step 2) else go to step 1).
2) Search all outgoing edges $e_1,\ldots,e_i$ of $m$ and take their destination nodes $d_1,\ldots,d_i$.
3) Adapt the node depths of the nodes $d_1,\ldots,d_i$ to depth $depth(m)+1$.
4) Add the nodes $d_1,\ldots,d_i$ to the end of the list $l$.
5) Return to step 1).

As visible, the minimum extraction is realized by taking the first node of the linked list. However, how does it work?

Since every edge in the graph is weighted with 1, every step in this Dijkstra algorithm insures that the depth of the node $d_j$ is greater - in case of $d_i$ - or equal - in case of $d_2,\ldots,d_i$ - to the previous entry in the list. The fact that every edge in the graph is weighted with 1 ensures that the minimum of all nodes is always at the beginning of the list. That means we only have to take the first element of the list to extract the node with minimum depth. The benefit of these modifications, we are now able to reduce the time to $O(1)$ for the insertion and $O(1)$ for the deletion.
In order to decide whether a node is visited or not, we consider the depth of a node. If it is smaller than the initial value - which was a very large integer \(\text{MAX\_INT}\) - we know that we have visited it before and drop it from the list.

The skipping of the binding between the Wikipedia level and the WordNet level (see Section 2.2.2) is realized in step 2) of the former listing. If the edge type is \textit{subclassof} and the destination node is a Wikipedia category, we take the destination node and repeat the step 2) with this destination node as source node.

All such Wikipedia category nodes get the depth minus one.

### 3.2 The Configuration File

In this section we explain the extensions we have done in the configuration file "topx_config.xml".

#### 3.2.1. Connection Data for the Ontology Database

In the old TopX environment, the corpus data and the ontology data were placed on the same database. This has changed for the YAGO ontology, which is placed on another database. In order to make TopX more flexible we displaced the connection data for the ontology database into the configuration file.

Thus we bring in the following new tags:

- \texttt{ONTOLOGY\_DB\_USERNAME}
- \texttt{ONTOLOGY\_DB\_USERPASSWORD}
- \texttt{ONTOLOGY\_DB\_HOST}
- \texttt{ONTOLOGY\_DB\_SID}

#### 3.2.2. Types of Relations

The relation types (i.e., the relation names) were hard coded in the old version of TopX. This is applicable if only few relation types exist and if the relation types change rarely - which was both the case at the WordNet ontology. With the YAGO ontology this changes. First of all YAGO has 96 different relation types and they could change with every new release of YAGO. This would denote that every changing of relation types in the YAGO ontology would cause a changing of the code, recompilation of the new code and updating of the code on the server. However, if the valid relation types were defined
in the configuration file, only this file would have to be updated. To this end, we changed the initialization of the relation types from hard coded to configuration file entries. We distinguish between the relation types from the WordNet ontology and the relation types from the YAGO ontology, so it is possible to switch between both ontologies.

Thus we introduced the following tags:

- RELATION_NAMES
- YAGO_RELATION_NAMES

Notice: Before someone changes the configuration file for the YAGO relation types he has to note of the entries of the table RELATION-NAMES in the database. If the relation types change in TopX for YAGO, it can not be avoided that the YAGO ontology has to be reimport completely. Afterwards the configuration file has to be modified as well.

### 3.2.3. Types of Relations for the Query Expansion

The query expansion does not work on all relation types, but on defined relation types. In the old TopX version, the respective relation types were hard coded in the code of the query expansion. To make this more flexible, we changed this allocation from hard coded to configuration file entries.

Thus we established the following new tags:

- ENABLED_EXPANSIONS
- YAGO_ENABLED_EXPANSIONS

The first one is for the WordNet ontology; the second one is for the YAGO ontology.

### 3.2.4. Switching between YAGO and WordNet

The following tag enables switching between the YAGO ontology and the WordNet ontology. We implemented the code such that if this flag is true, all procedures for the YAGO ontology are automatically used. Otherwise, all procedures for the WordNet ontology are automatically used.

- IS_YAGO_ONTOLOGY
Chapter 3 Changes for the New YAGO Version

3.3 Extensions for the YAGO Ontology

In the following sections we explain the extensions we have done for the YAGO ontology.

3.3.1. Import of Value Relations

As in Section 2.3.4 already mentioned, value relations like bornondate were not imported in thesis [18]. However, we think that it might enhance the results of the data retrieval if we are able to expand such values (e.g., dates, numbers, and other literals which are not words) to the query terms (see also query expansion on Section 2.3.2).

Here is an example:

We have the query "Albert Einstein" and could be able to add the value "14.03.1879", his date of birth, to the query terms. We obtain this value with the relation bornondate.

For this purpose we introduced a procedure which imports facts of defined value relations. This process exists of three steps:

1) First of all, all distinct values of the defined value relations, will be collected in the table VALUE_MEANS.
2) All collected values become concepts in TopX, there the value is the concept name and also the only word of the concept. That means for the upper example, if we find the value "14.03.1879" we add a concept with the name "14.03.1879" and his only word is "14.03.1879". This kind of concepts has no description.
3) After all concepts are generated for the collected values, the corresponding facts (i.e., edges in the ontology graph) will be included in the RELATIONS table.

In the current version of TopX all value relations are contained. If one value relation should be included during the query expansion, the respective name of the relation type has to be added in the configuration file (see also Section 3.2.3). Per default only the IS-A relations subclassof and type forward, subclassof and type backward, and ispartof are defined for the query expansion.
3.3.2. Two Kinds of Word Reducing

In TopX, words can be reduced to word stems or to unstemmed words. In both cases, stop words and special signs are dropped.

Here is an example:
If we have the word "things that matter" we have the following both reductions:

- Reduced to word stems: "thing matter".
- Reduced to unstemmed words: "things matter".

Which kind of reducing is used can be defined in the configuration file "topX_schemas.xml" via the tag \texttt{DO_STEM}. If this flag is true the words are reduced to their word stems; otherwise, they are reduced to unstemmed words.

In the table \texttt{SYNONYMS}, there was only one column served for the stemmed words. This means, that if the configuration for the tag \texttt{DO_STEM} changed, also the entries in the \texttt{SYNONYMS} table had to be replaced by the other reducing.

We added a further column to the table \texttt{SYNONYMS} for the second kind of reducing, so that the configuration can be changed and the code takes automatically the respective reducing out of the respective column. In that manner the \texttt{SYNONYMS} table does not have to be changed for such configuration changes.

3.3.3. Import of New Words

In case of YAGO provides new words, we do not want to have to reimport the complete YAGO ontology to obtain these new words.

Hence, we enabled the import of belated words. To this end, we provide the function "ImportWords".

In YAGO, words are assigned with the value relations \texttt{familynameof}, \texttt{givennameof}, \texttt{iscalled}, \texttt{isnativenameof} and \texttt{means}. Within this function, the names of the relation types (i.e., the names of the folders) - which contains the respective fact files – has to be defined. There exists the possibility to define files which should be ignored during the import. If entities will be found at the import, wherefore concepts not yet exist in the ontology, their concepts will be automatically created.
3.4 Statistics of the YAGO Ontology

To have present data for the similarity computation, we imported the newest version (March 2008). To have a general view we list the statistics of both versions, December 2007 and March 2008.

If some numbers are absence, this is due to the fact, that some of the calculations and code were not available at version December 2007.

Table 3-1

<table>
<thead>
<tr>
<th>CONCEPTS</th>
<th>Description</th>
<th>Version 12’2007</th>
<th>Version 03’2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>entries</td>
<td></td>
<td>1,477,641</td>
<td>2,160,771</td>
</tr>
<tr>
<td>WordNet concepts</td>
<td></td>
<td>69,108</td>
<td>69,108</td>
</tr>
<tr>
<td>concepts with descrip-</td>
<td></td>
<td>1,116,932</td>
<td>1,493,661</td>
</tr>
<tr>
<td></td>
<td>word</td>
<td>262</td>
<td>262</td>
</tr>
<tr>
<td>roots</td>
<td></td>
<td>533</td>
<td>533</td>
</tr>
<tr>
<td>roots without words</td>
<td></td>
<td>186</td>
<td>186</td>
</tr>
<tr>
<td>Wikipedia categories</td>
<td></td>
<td>152,342</td>
<td>175,716</td>
</tr>
<tr>
<td>value concepts</td>
<td></td>
<td>-</td>
<td>261,264</td>
</tr>
</tbody>
</table>

- It is observable that not all of the concepts have descriptions, although there exists a Wikipedia page for every concept. This is due to the following fact:
  For the import of the descriptions, we used a Wikipedia HTML dump. However, the latest version of the HTML dump which Wikipedia offered was the version of April 2007. Since this version is older than the Wikipedia version which YAGO used for his ontology, some concepts have not yet a Wikipedia page in that HTML dump and therefore also no description. Moreover, it has to be regarded that all value concepts have no description.
- The formulation "concepts without words" denotes that these concepts only appear in the table RELATIONS and have no entries in the tables CONCEPTS and SYNONYMS; only WordNet concepts are of this kind. This is because not all concepts from WordNet are available in the YAGO ontology as concepts, but can appear in so-called object relations (see also Section 2.2). The respective facts could be cut from TopX, in effect. But they help to define the correct depth for all other concepts, and therefore we keep these facts.
- The notation "Wikipedia categories" describes those concepts which are in the middle layer of the YAGO ontology. (see also Section 2.2.2)
3.4. Statistics of the YAGO Ontology

- The formulation "value concepts" signify those concepts which we obtain by so-called value relations (see also Section 2.2). Values are for example "01.01.2000" or "10 inch".

Table 3-2

<table>
<thead>
<tr>
<th>ENTITYIDS</th>
<th>Description</th>
<th>Version 12'2007</th>
<th>Version 03'2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>entries</td>
<td>1,408,533</td>
<td>2,091,678</td>
<td></td>
</tr>
<tr>
<td>value concepts</td>
<td>-</td>
<td>261,264</td>
<td></td>
</tr>
</tbody>
</table>

- Table 3-2 contains the mapping between the Wikipedia concept names and their concept IDs.

Table 3-3

<table>
<thead>
<tr>
<th>SYNONYMS</th>
<th>Description</th>
<th>Version 12'2007</th>
<th>Version 03'2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>entries</td>
<td>3,389,157</td>
<td>5,248,108</td>
<td></td>
</tr>
<tr>
<td>different words</td>
<td>2,336,774</td>
<td>4,092,674</td>
<td></td>
</tr>
<tr>
<td>different words (reduced to word stems)</td>
<td>2,237,572</td>
<td>3,796,314</td>
<td></td>
</tr>
<tr>
<td>different words (reduced to unstemmed words)</td>
<td>-</td>
<td>3,893,609</td>
<td></td>
</tr>
<tr>
<td>value words</td>
<td>-</td>
<td>261,264</td>
<td></td>
</tr>
</tbody>
</table>

- In this table is visible, that the number of words has doubled within the new version, while the count of concepts has risen minimally (see Table 3-1).

Table 3-4

<table>
<thead>
<tr>
<th>RELATIONS</th>
<th>Description</th>
<th>Version 12'2007</th>
<th>Version 03'2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of the relation types</td>
<td>-</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>number of the value relations entries</td>
<td>-</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>number of type facts</td>
<td>4,556,773</td>
<td>5,922,647</td>
<td></td>
</tr>
<tr>
<td>number of subclassof facts</td>
<td>3,306,172</td>
<td>4,474,808</td>
<td></td>
</tr>
<tr>
<td>number of type facts to Wikipedia</td>
<td>198,494</td>
<td>234,573</td>
<td></td>
</tr>
<tr>
<td>number of type facts to Wikipedia</td>
<td>2,914,009</td>
<td>3,947,690</td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 3 Changes for the New YAGO Version

<table>
<thead>
<tr>
<th>RELATIONS</th>
<th>Description</th>
<th>Version 12’2007</th>
<th>Version 03’2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>categories</td>
<td>number of <strong>subclassof</strong> facts in the WordNet level</td>
<td>58,857</td>
<td>58,857</td>
</tr>
<tr>
<td>number of value facts</td>
<td>-</td>
<td>661,482</td>
<td></td>
</tr>
</tbody>
</table>

- The notation "**type** facts to Wikipedia categories" match those facts, which connect the Wikipedia individuals with the Wikipedia categories (see also Section 2.2.2).
- The description "**subclassof** facts in the WordNet level" match the **hypernym** and **hyponym** relations out of WordNet.
Chapter 4

Similarity Measures

Similarity is a widely used concept. Most of the definitions of similarity depend on a particular application or a form of knowledge representation. There are many kinds of similarity measures, such as information content [8], mutual information [9], Dice coefficient [10], Cosine coefficient [10], distance-based measurements [11], and feature contrast model [12]. In this thesis we will only consider information content, Dice coefficient, and distance-based measurement, since these measures are applicable for the TopX application and its knowledge representation, the ontology.

Dekang Lin [13] listed three intuitions for similarity:

- **Intuition 1**: The similarity between $A$ and $B$ is related to their commonality. The more commonality they share, the more similar they are.
- **Intuition 2**: The similarity between $A$ and $B$ is related to the differences between them. The more differences they have, the less similar they are.
- **Intuition 3**: The maximum similarity between $A$ and $B$ is reached when $A$ and $B$ are identical, no matter how much commonality they share.

The similarity measures are applied on two words $a$ and $b$ from two distinct concepts $A$ and $B$ which are related through an edge in the ontology, where $a$ is a word of $A$ and $b$ is a word of $B$. We define the similarity value of two words $a$ and $b$ as $\text{sim}(a,b)$. The similarity value $\text{sim}(A,B)$ of two concepts $A$ and $B$ is based on the similarity values of their words; if one of these concepts has more than one word, then it gains the maximum of all similarity values of all possible words pairs between these two concepts.

$$\text{sim}(A,B) = \max_{i=1, j=1}^{n, m} \left[ \text{sim}(a_i, b_j) \right], \text{ where } a_i \in A, b_j \in B. n \text{ is the number of words of } A \text{ and } m \text{ is the number of words of } B. \quad (4.1)$$

The weight of the edge, connecting two concepts, is the similarity value of the respective concepts; all weights are saved in the table RELATIONS. The similarity value of concepts, which are related over a sequence of edges (i.e., path) will be computed during the query expansion (see also Section 2.3.2).
In this section we introduce all similarity measures which will be included for the similarity computation. For the frequency computation of the words we use a corpus, which is assembled for TopX by English documents.

### 4.1 Dice Similarity

This measure is the already used one in TopX.

#### 4.1.1. Default Definition

The Dice similarity measure, also called Dice coefficient [15], is as per Dekang Lin [13] one of the most commonly used similarity measure. As described in [14], the Dice coefficient measures the spatial overlap between two segmentations, \( R_1 \) and \( R_2 \) target regions, and is defined as follows:

\[
DC(R_1, R_2) = \frac{2|R_1 \cap R_2|}{|R_1| + |R_2|}
\]  

(4.2)

where \( \cap \) is the intersection of two regions [16].

Figure 4-1 shows, the more both regions are intersecting each other the higher the value of \( DC(R_1, R_2) \).

![Figure 4-1](image-url)
4.1. Dice Similarity

4.1.2. Definition in TopX

The segmentation $R_1$ and $R_2$, respectively, is represented in TopX by the amount of documents where the word $a$ and the word $b$, respectively, is mentioned. Moreover, the intersection of the segmentations $R_1$ and $R_2$, is represented by the amount of documents where the words $a$ and $b$ are commonly mentioned.

Thus the formula for the Dice similarity is defined as follows in TopX:

$$sim(a,b) = DC(a,b) = \frac{2 \cdot (FREQ(a,b))}{FREQ(a) + FREQ(b)}$$  \hspace{1cm} (4.3)

$a$ and $b$ are words of two directly related concepts (i.e., connected through an edge). $FREQ(a,b)$ is the number of documents where $a$ and $b$ are mentioned together; $FREQ(a)$ is the number of documents where $a$ is mentioned; $FREQ(b)$ is the number of documents where $b$ is mentioned.

The Dice similarity of the respective concepts $A$ and $B$ is defined as follows:

$$sim(A,B) = DC(A,B) = \max_{i,j=1}^{n,m} [DC(a_i,b_j)], \text{ where } a_i \in A, b_j \in B \text{ are the words.}$$  \hspace{1cm} (4.4)

For two distinct concepts $C_1$ and $C_{n+1}$, which are related over one unique relation path $p = \langle C_1, ..., C_{n+1} \rangle$ with length $n$, the Dice similarity is defined by the product of the similarity values of each relation $r_1, ..., r_n$ in this path $p$:

$$DC(p) = \prod_{j=1}^{n} DC(r_j), \text{ where } DC(r_j) = DC(C_j, C_{j+1})$$  \hspace{1cm} (4.5)

$DC(p)$ is the Dice similarity of the path $p$, $DC(r_j)$ is the Dice similarity of relation $r_j$ (i.e., an edge in the ontology graph), where $1 \leq j \leq n$. $C_j$ and $C_{j+1}$ respectively, are the corresponding concepts of relation $r_j$. This similarity computation is based on the maximum likelihood of independent successive relations. Moreover, if path $p$ contains just one relation $r$ (i.e., the concepts are related through an edge), then $DC(p)$ is equal to $DC(r) = DC(C_1, C_2)$.

This similarity measure is independent of the visiting direction; hence, the computed weights for one edge are valid for the forward similarity and the backward similarity. That means the columns "forward_sim" and "backward_sim" of the table RELATIONS have the same entries.
4.2 NAGA Similarity

NAGA [17] is concerned with ranking answers to semantic queries. Since YAGO provides the database for NAGA, it is obvious that the NAGA measure belongs to the considered measures in this thesis.

4.2.1. Default Definition

The NAGA similarity measure is based on the following three estimations.

1) A fact that has a high extraction confidence from authoritative pages should get a higher score than a fact with a lower extraction confidence.

2) The more informative a fact is, the higher should the similarity be. For example, ((Albert Einstein), IS-A, (physicist)), should get a higher similarity than ((Albert Einstein), IS-A, (politician)), since Albert Einstein is rather known as a physicist than as a politician.

3) Directly related concepts should get a higher similarity than distantly related concepts.

Thus we have three different values, the confidence, the informativeness and the compactness. First of all, we start with the explanation of the computation of the confidence and informativeness. After that, we will show that the compactness is already included in the confidence and informativeness value.

The confidence of a relation \( r \) (i.e., a fact or an edge) depends on the estimated accuracy \( \text{acc}(r, w) \) with which the relation \( r \) was extracted from a witness \( w \), an URL in which the relation \( r \) occurred - and the trust \( \text{tr}(w) \) we have in \( w \). Suppose that the relation \( r \) was extracted from the witnesses \( w_1, \ldots, w_n \). The formula of the confidence is:

\[
\text{conf}(r) = \frac{1}{n} \sum_{i=1}^{n} \text{acc}(r, w_i) \cdot \text{tr}(w_i)
\]  

(4.6)

The accuracy value is usually provided by the extraction mechanism; the trust in a witness can be computed by any algorithm similar to "PageRank" [24].

The confidence of a relation path \( p \) with length \( n \) is defined by the product over the confidence values of each relation in the path \( p = \langle r_1, \ldots, r_n \rangle \):

\[
\text{conf}(p) = \prod_{j=1}^{n} \text{conf}(r_j)
\]  

(4.7)
\( \text{conf}(p) \) is the confidence of the path \( p \), \( \text{conf}(r_j) \) is the confidence of relation \( r_j \), where \( 1 \leq j \leq n \).

The informativeness of a relation \( r = (x, t, y) \) depends on the unbound arguments of the relation. \( x \) and \( y \) are words out of the two corresponding concepts and \( t \) is the relation type.

The formula of the informativeness is defined as follows:

\[
\inf(r) = P(x \mid t, y), \text{ if } x \text{ unbound in } r
\]

and

\[
\inf(r) = P(y \mid t, x), \text{ if } y \text{ unbound in } r
\]

Unbound means that \( x \) and \( t \) are given, and we search for all possible \( y \) (i.e., \( y \) is unbound).

As an example, we search for all vocations of "Albert Einstein"; this yields to the relation \( r = (\text{Albert Einstein}, \text{IS-A}, y) \).

\( P(x \mid t, y) \) can be written as follows:

\[
P(x \mid t, y) = \frac{P(x, t, y)}{P(t, y)}
\]

where \( P(x, t, y) \) is the number of witness pages for the relation \( r \).

In [17] they estimated

\[
P(x \mid t, y) \sim P(x \mid y)
\]

hence

\[
P(x \mid t, y) \sim \frac{P(x, y)}{P(y)}
\]

and

\[
P(y \mid t, x) \sim \frac{P(x, y)}{P(x)}
\]

With the prior example we get the formula \( \inf(r) = P(y \mid \text{IS-A, Albert Einstein}) \). If we search for the informativeness of the fact that Albert Einstein is a politician, we obtain the informativeness

\[
\frac{P(\text{Albert Einstein, politician})}{P(\text{Albert Einstein})},
\]

where \( P(\text{Albert Einstein, politician}) \) is the number of pages containing both words and \( P(\text{Albert Einstein}) \) is the number of pages containing "Albert Einstein".
The informativeness of a path $p$ with length $n$ is defined by the product over the informativeness values of each relation in the path $p = \langle r_1, \ldots, r_n \rangle$:

$$\inf(p) = \prod_{j=1}^{n} \inf(r_j) \quad (4.14)$$

$\inf(p)$ is the informativeness of the path $p$, $\inf(r_j)$ is the informativeness of relation $r_j$, where $1 \leq j \leq n$.

As visible above, the formula of the confidence and the informativeness for a relation path $p$ is the product over the values of its component relations. That means, the more relations in a path exist the lower is the resulting value of the confidence and the informativeness. That way, the compactness is taken into account as well.

The final NAGA similarity measure is a combination of the confidence, the informativeness and the compactness - where the compactness is included in the confidence and the informativeness formulas. This measure is applicable on a single relation or on a complete path of relations as well, and is defined as follows:

$$NAGA(s) = \beta \cdot \text{conf}(s) + (1 - \beta) \cdot \inf(s) \quad (4.15)$$

where $0 \leq \beta \leq 1$; $s$ can be a single relation or a path of relations.

If $s$ is a single relation $r$, $\text{conf}(r)$ is the confidence of the relation $r$ and $\inf(r)$ is the informativeness of the relation $r$.

If $s$ is a path of relations $r_1, \ldots, r_n$, $\text{conf}(s)$ is the product of the confidence values of the single relations $r_j$ on the path $s$ and $\inf(s)$ is the product of the informativeness values of the single relations $r_j$, where $1 \leq j \leq n$.

### 4.2.2. Definition in TopX

First of all we consider the informativeness value. We adopt $P(x)$ and $P(y)$ as the number of documents where $x$ respectively $y$ is mentioned and $P(x, y)$ as the number of documents where $x$ and $y$ are commonly mentioned.

Thus the informativeness for each word pair $(x, y)$ out of the relation $A \rightarrow B$ is defined as follows in TopX:

$$\inf(x, y) = \frac{\text{FREQ}(x, y)}{\text{FREQ}(x)}, \quad (4.16)$$
and for each word pair \((y, x)\) out of the relation \(B \rightarrow A\):

\[
\inf(y, x) = \frac{FREQ(x, y)}{FREQ(y)},
\]

(4.17)

where \(x \in A, y \in B\).

For the purposes of TopX we need the informativeness of two directly related concepts. We define this informativeness as follows:

\[
\inf(A, B) = \max_{i,j \in 1} \left[ \inf(a_i, b_j) \right], \text{ where } a_i \in A, b_j \in B \text{ are the words.}
\]

(4.18)

We store the informativeness values in the \textbf{RELATIONS} table. As visible in the formulas before, the informativeness values are dependent on the visiting direction. That means the columns "forward_sim" and "backward_sim" of the table \textbf{RELATIONS} have different entries.

Finally, we regard the confidence value. In YAGO, all confidence values were already computed and stored in a column in the fact files. To make them more accessible for TopX, we stored them in the database. The confidence values could be stored in a further column in the \textbf{RELATIONS} table; however, the confidence is a special part of the NAGA measure only; therefore, instead of saving this information in the \textbf{RELATIONS} table we introduced a new table \textbf{RELATIONCONFIDENCES}. To relocate the confidence values to the correct relations, we also stored the source concept ID, the target concept ID, and the relation ID with each confidence value.

<table>
<thead>
<tr>
<th>RELATIONCONFIDENCES</th>
<th>Description</th>
<th>Version 03’2008 entries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5,922,781</td>
</tr>
</tbody>
</table>

The final NAGA similarity measure is defined as follows:

\[
sim(A, B) = NAGA(A, B) = \beta \cdot conf(A, B) + (1 - \beta) \cdot \inf(A, B).
\]

As already mentioned in the default definition, the concepts \(A\) and \(B\) can be related over a single relation or over a path of relations. The similarity values of concepts which are related through a path are computed during the query expansion (see also Section 2.3.2); however, since the NAGA similarity measure is a summation of two parts, the path similarity computation has to be adopted. This adopting will be explained in the Section 7.2.
4.3 Best WordNet Similarity

For WordNet there exist a lot of measures to determine the degree of semantic similarity between two distinct concepts. In this section we explain the measure [19], which achieved best results in comparison to other WordNet measures [8], [13], [14], [21], [22], until 2007.

4.3.1. Default Definition

*Yuhua Li at al.* [19] found the best measure for similarity values between words inside the WordNet ontology. In the paper [19] they adopt a number of information sources, which consist of structural semantic information from a *lexical taxonomy* (i.e., ontology) and information content from a corpus. They implemented a variety of strategies for using various possible information sources, to investigate how information sources could be used effectively. We introduce here that strategy which yielded the best result. The best strategy was the combination of the following structural semantic information:

- Length of paths between two concepts.
- Abstractness of a concept.

Both were retrieved from the lexical taxonomy WordNet. To achieve the similarity between two words, we regard the concepts in the ontology, which contain the compared words. Before we continue with explaining how these information will be obtained from the ontology, we introduce the term *subsumer*. A subsumer is the concept which subsumes the regarded concepts. In this measure always the first subsumer will be considered.
4.3. Best WordNet Similarity

We show an excerpt of the example of the hierarchical semantic knowledge base in [20]:

In Figure 4-2 the concept "person, human, …" - illustrated by the brighter circle - is the first concept which subsumes the concepts "teacher, instructor" and "male child, boy, child", and therefore the considered subsumer. The upper concept "life form, being, …" is not the considered subsumer, since it is not the first concept which subsumes the other concepts.

The first semantic information we regard is the length of paths between two concepts. The idea behind this approach is the shorter the path between two concepts, the higher the similarity of these concepts. Thereby those paths will be considered, which go through the first subsumer. If there exist multiple paths between two concepts, they take only the shortest paths into account. This method for similarity calculation is also mentioned in the approach of Rada et al. [20]. This means in the upper picture, that the concept "adult, grownup" is more similar to the concept "male, male person" than to the concept "male child, boy, child".

But this sole issue may be not so accurate if it is applied to larger and more general semantic nets such as WordNet. As an example, the minimum length from the concept "male, male person" to the concept "animal, beast, …" is 3 and the minimum length from the concept "male, male person" to "teacher, instructor" is 5. In this case the first path length is less than the second one, but we would not say "male person" is more similar to "animal" than to "teacher".

Due to this incorrectness they introduced the second semantic information abstractness of a concept. The abstractness of a concept is mapped to the depth of the concept in the hierarchy. They established that concepts at upper layers
in WordNet have more general semantics, for example the concept "entity", and therefore less similarity between them.

The final similarity between two concepts is defined as follows:

\[ \text{sim}(A, B) = f_1(l_1) \cdot f_2(h_2) \]

where \( f_1 \) and \( f_2 \) are transfer functions of path length and abstractness / concept depth, respectively. \( (4.19) \)

The similarity is set to 1 if the concepts have exactly the same meaning and is set to 0 for no similarity. That means the interval of the similarity is \([0, 1]\).

**The Transfer Function of the Path Length**

The path length \( l \) is applied on the concepts which contain the compared words. They differ between three cases:

- Both words are in the same concept: These words have the same meaning \( \Rightarrow \) they attach the path length 0.
- Both words are in different concepts but these concepts share more than one common word: These words partially share the same features \( \Rightarrow \) they assign the path length 1.
- Both words are in different concepts and these concepts share no common words: These words are most different \( \Rightarrow \) they allocate the actual minimal path length between the concepts.

The transfer function of the path length \( l \) is defined as follows:

\[ f_1(l) = e^{-\alpha l} \]  \( (4.20) \)

\( f_1 \) is a monotonically decreasing function of \( l \), where \( \alpha \) is a constant. It satisfies the constraint that the similarity should have values in the interval of \([0, 1]\).

**The Transfer Function of the Abstractness / Concept Depth**

For this issue they pick the depth of the first subsumer of the considered concepts. The depth \( h \) of a concept is obtained by counting the levels from the concept to the top of the lexical hierarchy. If there exist multiple paths between two concepts, the first subsumer on the shortest path will be selected.

They defined the transfer function of the first subsumer depth \( h \) by the following formula:

\[ f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \]  \( (4.21) \)
4.3. Best WordNet Similarity

$f_2$ is a monotonically increasing function of $h$, where $\beta > 0$ is a smoothing factor. As $\beta \to \infty$, the depth of a concept is not considered. It satisfies also the constraint, that the similarity should have values in the interval of $[0, 1]$.

**Final Similarity Measure**

The Best WordNet similarity measure is the product of the transfer function of the path length and the transfer function of the abstractness.

The formula is defined as follows:

$$WordNet(A, B) = f_1(l) \cdot f_2(h)$$  \hspace{1cm} (4.22)

hence,

$$WordNet(A, B) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$  \hspace{1cm} (4.23)

where $\alpha \geq 0$ and $\beta > 0$ are parameters scaling the contribution of shortest path length and depth, respectively.

4.3.2. Definition in TopX

At first we take a look at the implementation of the path length computation. We have to denote that the path length $l$ in this measure does not correspond to the path length $k$ in TopX. The path length in this measure, is the path length in the IS-A hierarchy of the ontology. That means in the YAGO ontology, if we have a relation (i.e., an edge) which is not an IS-A relation, we have to take the path length $l$ of the associated concepts in the IS-A hierarchy, whereas the path length $k$ is always equal to one, since we consider edges. That means if the type of the relation equates to IS-A relation, the corresponding path length $l$ is also one; otherwise, the corresponding path length $l$ is always greater than one.
In Figure 4-3 the black arrows denote the IS-A relations. The red arrow - labeled with *born in* - pictures the relation for which the similarity has to be computed and the red points are the associated concepts. Although the relation ("Albert Einstein", *born in*, "Ulm") has length 1 the associated path length in the IS-A hierarchy is 5.

Next, we consider the *super concepts*. The super concepts of a concept are those concepts which can be passed by a directed walk through the IS-A hierarchy from the concept to the root.

To achieve the minimal path length of two concepts \( C_1 \) and \( C_2 \) in the IS-A hierarchy, first of all, we collect all super concepts \( S_{1,1}, \ldots, S_{1,n} \) and \( S_{2,1}, \ldots, S_{2,m} \) of both concepts separately, with a breadth first search on the IS-A graph from the respective concept up to the source of the IS-A hierarchy. Next we compare both lists \( S_{1,1}, \ldots, S_{1,n} \) and \( S_{2,1}, \ldots, S_{2,m} \). Since both lists are filled in a breadth first order, the first common super concept \( S = S_{1,j} \) or \( S = S_{2,j} \) with \( S_{1,j} = S_{2,j} \) - from the beginning of the lists (i.e., from position 1) - in those lists is the first subsumer of both concepts. During this collection of the super concepts for a concept, also their distances \( d \) from this concept are listed. That means an entry of such a list is \( S_j = (C,d) \), where \( c \) is the found concept and \( d \) is the distance from the started concept.

Thus we only have to add the distances of the first subsumer in both lists to achieve the minimal path length.
Figure 4-4 shows an example:

In Figure 4-4, the list of the super concepts for the concept $A$ in breadth first order is $[(C,1), (D,1), (S,2), (H,2), (S,3)]$ and for concept $B$ it is $[(F,1), (G,1), (E,2), (S,2), (H,3), (S,4)]$. The first common concept (i.e., from the left) in both lists - and therefore the first subsumer - is $S$; the according path length is $2 + 2 = 4$. $H$ also is a subsumer of both, but - since it appears in the super concepts list of concept $B$ after the entry $(S,2)$ - not the first subsumer.

To make these values - the first subsumer and the minimal path lengths – in perpetuity accessible for TopX, we stored them in the database. For this purpose we introduced the new table `RELATIONINFOS`. To allocate the values to the respective relation, we also store the respective source concept ID, target concept ID and relation ID.

In Table 4-2 we show an overview of the computed path length values, in the IS-A hierarchy, for the edges.
Moreover, path lengths of 0 will never occur, since we only compute the similarity value of two distinct concepts, which are related over a single edge or a sequence of edges (i.e., path).

Next we regard the depth of the concepts. As already mentioned in Section 3.1, we use the Dijkstra algorithm for the computation of each concepts depth $h$. In the old TopX the Dijkstra algorithm was only used for the computation of the environment of a concept for the word sense disambiguation. For this computation the graph with forward and backward edges of each IS-A relation was needed. On behalf of the depth computation we need only the backward edges of the IS-A relations (i.e., edges from the root node to the concepts). Hence we implemented a second graph representation which can be used for the depth computation. We have to note that the YAGO ontology has more than one root. To avoid the processing of the Dijkstra algorithm on all root nodes and comparing the computed values with precomputed values from other root nodes, we expand the graph representation as follows. We add an auxiliary node which illustrates the unique root and add auxiliary edges from this unique root to the real roots. After that, the graph representation contains one single root node and the Dijkstra algorithm can be started on it (see Figure 4-5).
To handle the auxiliary edges, we subtract from every depth the value one. Thereby we get depth 0 for the real root nodes and the real depth for the remaining concepts.

To have an idea how dense and how broad the YAGO ontology is, we show a short overview of the depth values:

<table>
<thead>
<tr>
<th>CONCEPTDEPTHS</th>
<th>Description</th>
<th>Version 03’2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>concepts with depth</td>
<td></td>
<td>2,160,771</td>
</tr>
<tr>
<td>depth = -1</td>
<td></td>
<td>272,495</td>
</tr>
<tr>
<td>depth = 0</td>
<td></td>
<td>357</td>
</tr>
<tr>
<td>depth = 1</td>
<td></td>
<td>1,171,304</td>
</tr>
<tr>
<td>depth = 2</td>
<td></td>
<td>11,673</td>
</tr>
<tr>
<td>depth &gt; 2</td>
<td></td>
<td>529,226</td>
</tr>
</tbody>
</table>

As already mentioned above, the similarity values are valid for a single edge of the ontology, where the path length is the corresponding path length of the respective concepts in the IS-A hierarchy (see also Figure 4-3). We define this similarity measure for concepts which are related through a path (i.e., a sequence of edges) in the ontology (i.e., not a path in the IS-A hierarchy), by the maximum likelihood of independent successive relations - as already used in the Dice similarity measure.
Chapter 4 Similarity Measures

The WordNet similarity of a relation path $p = \langle C_1, ..., C_{n+1} \rangle$ with length $n$ is defined by the product over the WordNet similarities of each relation $r_1, ..., r_n$ in the path:

$$WordNet(p) = \prod_{j=1}^{n} WordNet(r_j), \quad (4.24)$$

where

$$WordNet(r_j) = WordNet(C_j, C_{j+1}) = e^{-\alpha \cdot \beta} \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}}. \quad (4.25)$$

$WordNet(p)$ is the WordNet similarity of the path $p$, $WordNet(r_j)$ is the WordNet similarity of relation $r_j$ where $1 \leq j \leq n$. $C_j$ and $C_{j+1}$ are the respective concepts of the relation $r_j$. If path $p$ contains just one relation $r$, then $WordNet(p)$ is equal to $WordNet(r)$.

We use for $\alpha$ and $\beta$ the values which achieved best results in the paper [19]: $\alpha = 0.2$ and $\beta = 0.6$.

This similarity measure is also independent of the visiting direction; hence, the computed weights for one edge are valid for the forward similarity and the backward similarity. That means the columns "forward_sim" and "backward_sim" of the table RELATIONS have the same entries.

4.4 Worked Samples

Let us consider the relation $r = (Albert\_Einstein, Is\_A, physicist)$. The concept Albert\_Einstein contains 23 words {"albert einstein", "albert", "einstein", ...} and the concept physicist contains only one word{"physicist"}.

For this relation we obtain the following similarity values.

**Dice Similarity**

The formula of the Dice similarity measure is defined as follows (see Section 4.1),

$$DC(A, B) = \frac{n_{\text{shared}}}{\frac{1}{2}n_{\text{A}} + \frac{1}{2}n_{\text{B}}} \max_{a_i, b_j} DC(a_i, b_j),$$

where $a_i \in A, b_j \in B$ are the words, and
4.4. Worked Samples

\[ DC(a,b) = \frac{2 \cdot (\text{FREQ}(a,b))}{\text{FREQ}(a) + \text{FREQ}(b)}. \]

The words which present the maximum similarity value for the Dice similarity are "einstein" from \textit{Albert Einstein} and "physicist" from \textit{physicist}:

\[
DC(\text{\textit{Albert Einstein, physicist}}) = DC(\text{"einstein", "physicist"})
\]

\[
= \frac{2 \cdot (\text{FREQ("einstein","physicist")})}{\text{FREQ("einstein")} + \text{FREQ("physicist")}} = \frac{2 \cdot 512}{2,829 + 6,262}
\]

\[= 0.1124 \]

\textbf{NAGA Similarity}

The formula of the NAGA similarity measure is defined as follows (see Section 4.2),

\[ NAGA(A, B) = \beta \cdot \text{conf}(A, B) + (1 - \beta) \cdot \inf(A, B). \]

The confidence value for the relation \( r \) is 0.9221. Since the NAGA similarity measure depends on the visiting direction we consider the similarity value of both directions. The informativeness is defined as follows,

\[ \inf(A, B) = \max_{i=1,j=1}^{n,m} \left[ \inf(a_i,b_j) \right], \] where \( a_i \in A, b_j \in B \) are the words, and

\[ \inf(x,y) = \frac{\text{FREQ}(x,y)}{\text{FREQ}(x)}, \] for the forward direction, and

\[ \inf(y,x) = \frac{\text{FREQ}(x,y)}{\text{FREQ}(y)}, \] for the backward direction.

The words with maximum similarity value for the forward direction are "albert einsteins" from \textit{Albert Einstein} and "physicist" from \textit{physicist}:

\[ \inf(\text{\textit{Albert Einstein, physicist}}) = \inf("albert einsteins", "physicist") \]

\[
= \frac{\text{FREQ("albert einsteins","physicist")}}{\text{FREQ("albert einsteins")}} = \frac{66}{237}
\]

\[= 0.2784, \]

and the words with maximum similarity value for the backward direction are "albert" from \textit{Albert Einstein} and "physicist" from \textit{physicist}:

\[ \inf(\text{\textit{Albert Einstein, physicist}}) = \inf("albert", "physicist") \]

\[
= \frac{\text{FREQ("albert","physicist")}}{\text{FREQ("physicist")}} = \frac{892}{6,262}
\]
For the final similarity measure we set $\beta = 0.5$:

$$NAGA(Albert \_ Einstein, Physicist) = 0.5 \cdot 0.922117 + 0.5 \cdot 0.2784$$

$= 0.6003$ for the forward direction, and

$$NAGA(Albert \_ Einstein, Physicist) = 0.5 \cdot 0.922117 + 0.5 \cdot 0.1424$$

$= 0.5323$ for the backward direction.

**Best WordNet Similarity**

The formula of the Best WordNet similarity measure is defined as follows (see Section 4.3),

$$WordNet(A, B) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}.$$  

The path length of the relation $r$ is 1 and the depth of its subsumer (i.e., of the concept $\text{physicist}$) in the IS-A hierarchy of YAGO is 5.

We set $\alpha = 0.2$ and $\beta = 0.6$ (best values in paper [19]) and obtain the similarity value:

$$WordNet(Albert \_ Einstein, \text{physicist}) = e^{-0.21} \cdot \frac{e^{0.65} - e^{-0.65}}{e^{0.65} + e^{-0.65}}$$

$= 0.8147$

### 4.5 Exceptions in the Similarity Computation

There exist two special treatments for the similarity computation. The first one is for an amount of edges which are concerned in the binding between the WordNet level and the Wikipedia level (see also Section 2.2.2) - a part of the type and subclassof edges. The second is for the value relations. A further issue, which requires an exception in the similarity computation, is the appearance of special signs within some words in the YAGO ontology.

These exceptions are independent of the similarity measures. That means that we have to care about them in all three measures.
4.5. Exceptions in the Similarity Computation

4.5.1. The Binding between YAGO and WordNet

Since the binding between the Wikipedia level and the WordNet level will always be skipped, the computation of the similarities has to be adapted to this exception. To this end we take a closer look to this binding.

For each Wikipedia individual in the Wikipedia level, there exist at least one type edge to the Wikipedia categories - the middle layer in the Wikipedia level (see also Section 2.2.2); whereas from the middle layer for each Wikipedia category there exists exactly one subclassof edge to the WordNet level. This fact is also visible in Figure 4-6. That denotes that we can take these type edges as valid edges for the similarity computation and leave the following subclassof edges.

In the default case we save the similarity values at the respective edge. Due to the fact that we visit two edges during the skipping of the middle layer, we have to decide at which edge we save the computed similarity value. Intuitively the concerning type edges are the correct candidates, so that the similarity value can be assigned unambiguously - since for each type edge it is clear which Wikipedia concept and which WordNet concept is affected. As the path similarity computation is realized by the product of the similarity values of the single edges we give the following subclassof edge the similarity 1, so that the resulting similarity stays unaffected.

We take a look at a path $p$ through the binding. This path consists of two edges, one is the type edge and the other is the subclassof edge. Due to the definition of the similarity measure, we obtain the following similarity formula for the Dice similarity and the WordNet similarity:

$$\text{sim}(p) = \text{sim(type)} \cdot \text{sim(subclassof)}$$

hence

$$\text{sim}(p) = \text{sim(type)} \cdot 1 = \text{sim(type)}$$

The respective formula for the NAGA similarity is:
\[ \text{sim}(p) = \beta \cdot (\text{inf}(\text{type}) \cdot \text{inf}(\text{subclassof})) + (1 - \beta) \cdot (\text{conf}(\text{type}) \cdot \text{conf}(\text{subclassof})) \]

hence

\[ \text{sim}(p) = \beta \cdot (\text{inf}(\text{type}) \cdot 1) + (1 - \beta) \cdot (\text{conf}(\text{type}) \cdot \text{conf}(\text{subclassof})) \]

For the NAGA similarity is to regard, that the unaffected part is the informativeness. The confidence of this path has to stay the product of the confidence of the type edge and the subclassof edge.

On each subclassof edge which is not included in the binding - inside the WordNet level - we compute and save the similarity values like in the default case.

### 4.5.2. Value Relations

Since there only exist a very small number of facts for each concept of one value relation type - in most cases even only one fact, as for example bornon-date - , and due to the fact that all these facts should have the same importance, we decided to give all value facts the similarity value one. That way, we handle all values as synonymous words. The backward similarity of each value relation will be set to zero, since we will not expand in that direction.

### 4.5.3. Special Signs

The Wikipedia concepts contain also words like different notations. That means we partially have Arabic characters, Chinese characters, and many more. During the frequency computation we realized that database requests on the corpus, for words which contained such characters, yields to a huge responding time, which was not reasonable. Therefore we ignored those words which contain characters with character code greater than 255 during the similarity computation. Since all words which contain characters with character code greater than 255 should not be available in the corpus anyway, this method should not affect the values of the similarity computation.
4.6 Efficient Computation

For the similarity computation were around 30,000,000,000 calculations to be done - since all pairs of words of one relation have to be considered - , and around 5,000,000 values have to be communicated with the database. This means a lot of time. The sequential computation for only one measure would take approximately over one week.

A possible speed up is the parallel computation. For this purpose we implemented the code for the similarity computation that way that it can be run in parallel or sequentially. The parallelism is realized with different parameters for the similarity function, where every measure has its own function.

The first kind of parallelism is to compute the three measures in parallel. If each function gets no parameters it computes automatically all relations with the respective similarity measure. We only have to start all three similarity functions in parallel.

The second kind of parallelism is to compute relations for one measure in parallel. To this manner each function has two parameters, which has to be used alternatively:

- Relation ID: If we commit the relation ID to the function, we can start the function in parallel for multiple types of relations.
- Query string on relations: If we assign a query string on relations to the function, we are able to start the function for multiple queries in parallel. These queries can be so detailed as we want (e.g., "SELECT * from RELATIONS where relationid = 77 and sourceconceptid >= 1000700000 and sourceconceptid < 1001050000").

That means we are able to compute the measure in parallel in different ranks and as detailed as we want; in the level of the measures (i.e., only the measures in parallel), in the level of relation types (i.e., in parallel for each measure), or in the level of queries (i.e., the most detailed case, which could be overall several relation types).
4.7 Overview

With the aid of the parallel computation we could calculate all similarity values for the three different measures in maximum around 4 days.

For comparison we show here an overview of the three different similarity values.

Table 4-4

<table>
<thead>
<tr>
<th>Similarities</th>
<th>Description</th>
<th>Version 12’2007</th>
<th>Version 03’2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dice &lt;&gt; 0</td>
<td></td>
<td>4,014,042</td>
<td>5,559,793</td>
</tr>
<tr>
<td>NAGA &lt;&gt; 0</td>
<td></td>
<td>4,014,942</td>
<td>5,559,793</td>
</tr>
<tr>
<td>Best WordNet &lt;&gt; 0</td>
<td></td>
<td>3,901,630</td>
<td>5,393,492</td>
</tr>
</tbody>
</table>

- Table 4-4 shows that only about 6 percent of all relations have a similarity equal to zero.

The values for each similarity measure are stored in different tables.

Notice: If we remind, in TopX, it is possible to reduce words to word stems or to unstemmed words (see also Section 3.3.2). We computed all similarity measures with the unstemmed reduced words of the concepts, since the configuration for DO_STEM of the available schemata is set on "false". If the configuration changes to "true", all similarity values have to be recomputed.
Chapter 5

Evaluation of the Computed Similarity Values

In this section we evaluate the computed similarity values.

Figure 5-1 shows a statistic distribution of the similarity values over $\delta$ for all measures. The legend of the x-axis describes values of $\delta$, where $\delta > 0$ means values between $0 < \delta \leq 0.01$ (e.g., the measuring point for the Best WordNet similarity on $\delta > 0.5$ shows, that around 85% of all relations have a similarity value between $0.5 < \delta < 1$). Dice (WordNet ontology) refers to the similarity values for the WordNet ontology and Best WordNet, NAGA backward, NAGA forward, and Dice refer to the similarity values for the YAGO ontology. Since the NAGA measure depends on the visiting direction, we consider both directions separately. It appears that the distributions of the Dice
similarity values (i.e., the green line) and the NAGA similarity values of the backward direction (i.e., the red line) between $0.01 < \delta \leq 1$ are close to the distribution of the similarity values for the WordNet ontology (i.e., the black line). Outliers are the NAGA similarity values of the forward direction (i.e., the orange line) and the Best WordNet values (i.e., the blue line). This could be evidence that these measures are not applicable for the YAGO ontology.

5.1 Best WordNet measure

A deeper look at the distribution of the Best WordNet values for $0.15.0 \leq \delta$ showed, that around 68% of all relations have a similarity value between $0.81 < \delta < 0.82$. This bundling is manifested due to the structure of the YAGO ontology.

The Best WordNet measure is the product of a transfer function of the path length and a transfer function of the concept depth of the subsumer of each relation (see Section 4.3). That means if the IS-A hierarchy of an ontology is a very dense graph (i.e., most of the relations have the same path length and most of their subsumers have the same depth), the resulting similarity values are also very close to each other.

If we consider the YAGO ontology, around 90% of all relations are IS-A relations and have therefore a path length of 1 (see also listing for the path lengths in Section 4.3.2). For these relations, the transfer function of the path length returns 0.8181. That means only the depth of their subsumer affects their final similarity value. Since around 76% of these relations have a subsumer whose depth is greater than 5, we obtain the distribution showed in Figure 5-1 of the similarity values for the Best WordNet measure.

These facts consolidate the assumption that this measure is not applicable for the query expansion on the YAGO ontology.

5.2 NAGA measure

The query expansion in TopX uses only defined relation types (see Section 2.3.2). These currently are hypernym, hyponym and part_meronym for the WordNet ontology and subclassof and type forward, subclassof and type backward, and ispartof forward for the YAGO ontology. In other words, the query expansion searches for more general and for more specific terms of the original query terms.

As seen in Figure 5-1 the second outlier was the forward direction of the NAGA measure, whereas the backward direction looks good.
Since 90% of all relations in YAGO are IS-A relations, we assume that the distribution of the forward direction in Figure 5-1 describes the distribution of the IS-A relations subclassof and type forward; in other words the direction to more general terms (e.g., "location" IS-A "object"). As already mentioned above, the query expansion actually also goes in the direction of more general terms of the source query terms. However, to search for more general terms is rather unusual in query expansion and it seems that the forward direction of the IS-A relations could be dropped from the query expansion without hesitation.

If the prior assumption holds, this would mean that the outlier of the forward direction is irrelevant for the quality of the NAGA measure.

To this end, we consider the similarity distribution especially for the relation types of the current query expansion (see listing at the beginning of this section 5.2).

![Relation Types of the Query Expansion](image)

**Figure 5-2**

Figure 5-2 shows, as already assumed, that the distribution for the relation types of the query expansion closely agrees with the distribution for all relation types. This fact is significant for the decision that the outlier of the forward direction of the NAGA measure is not problematic.

This agreement of Figure 5-1 and Figure 5-2 denotes that the relation types subclassof and type forward, could be dropped from the configuration for the query expansion and the fact that the forward direction of the NAGA measure is an outlier is not problematic for the quality of this measure. Furthermore, we
have seen in Figure 5-1 and Figure 5-2 that the more interesting backward direction closely corresponds to the distribution of the similarity values for the WordNet ontology.

**The Influence of the Confidence**

For the NAGA values in Figure 5-1 and Figure 5-2 only the informativeness values were consulted. However, the final NAGA value of a relation is a combination of the informativeness value and the confidence value (see Section 4.2):

\[ NAGA(A, B) = \beta \cdot conf(A, B) + (1 - \beta) \cdot \inf(A, B) \]

We want to show what happens with the distribution in Figure 5-1 and Figure 5-2, if the confidence value will be included.

We know that all confidence values are above 0.9. Let us consider the following example:

We include the confidence values and set \( \beta = 0.5 \), so we obtain the following similarity values:

\[ NAGA(A, B) \geq 0.45 + 0.5 \cdot \inf(r) \]

With this example we can see that the higher \( \beta \) is, the higher is the final similarity value, since the confidence values are above 0.9. Furthermore, we can see that the higher \( \beta \) is, the smaller is the part of the informativeness.

These two observations lead to a compression of the distribution graph for the NAGA measure in Figure 5-1 and Figure 5-2, in the area of \( \delta \) in the interval \( [\beta \cdot 0.9, \beta \cdot 0.9 + (1 - \beta) \cdot \inf(A, B)] \).

![Figure 5-3](image)

Figure 5-3 demonstrates the compression of the NAGA backward graph for \( \beta = 0.1 \) and \( \beta = 0.5 \).
As also visible in Figure 5-3, the left side of the distribution graph can be compressed up to $\delta = 0.9$ (i.e., in an area about 0.9), whereas the right side can be only compressed down to 0.9 (i.e., in an area about only 0.1), conditioned on the choice of $\beta$.

This compression is significant for the query expansion. Since it has to be regarded, that the choice of $\delta$ has to be adapted to the choice of $\beta$ (i.e., the size of the influence of the confidence values).

Concluding, we can say that the NAGA measure is applicable for the query expansion, with the restrictions of excluding of the forward direction and regarding on the choice of $\delta$. 
Chapter 6

Word Sense Disambiguation at the YAGO ontology

The application of the word sense disambiguation as defined for the WordNet ontology (see also Section 2.3.1) posed a problem on the YAGO ontology. In this section we will explain what the problem is and list some resolution methods. Moreover, we show the evaluation of several tests with these resolution methods.

6.1 The Problem

First of all, we do a short review to a part of the word sense disambiguation (see also Section 2.3.1). The word sense disambiguation assigns to each candidate concept $s_j$ the respective local context $\text{con}(s_j)$. This local context includes the own words of the candidate concept, and the words of all respective immediate hypernyms and hyponyms up to a depth of two including their descriptions.

This approach was fine for a smaller ontology like WordNet. The problem on the YAGO ontology consists in a large linked graph and the environment of a concept within YAGO, which could include more than 100,000 concepts. For this reason, we have to change the approach of the context extraction in the word sense disambiguation, to make it applicable for the YAGO ontology; otherwise it would take too much time and a lot of memory capacity.

Figure 6-1 depicts a possible environment of a concept in the YAGO ontology. The dashed arrows illustrate the high linked binding between the WordNet level and the Wikipedia level (see also Section 2.2.2).
6.2 Methods of Resolution

6.2.1. WordNet Senses

We start with the consideration of the WordNet senses (i.e., all concepts in the WordNet level). Since we are able to separate the WordNet concepts from the Wikipedia concepts (we stored with each concept its provenance), we leave the approach of the local context allocation for WordNet concepts as defined; however we add the condition that only WordNet concepts belong to the environment of a WordNet concept. In other words, we stay inside the upper area within Figure 6-1. In doing so, the highly linked binding between the WordNet level and the Wikipedia level will be ignored, ensuring that the local context of such concepts is as good as in the old approach.

6.2.2. Wikipedia Senses

Further we consider the Wikipedia senses (i.e., all concepts in the Wikipedia level). First of all, we tried to take the own words and the own description of the Wikipedia concepts as local context only. Since the description of a Wikipedia concept is the content of the respective Wikipedia page, the description and the own words should contain enough information to be adequate as local context. However, tests with this local context revealed that the Wikipe-
dia pages have to much irrelevant words; so that also concepts, which have nothing in common with the local context of the considered term (see Section 2.3.1), get a similarity value above the similarity values of real relevant concepts. As an example we consider the query "U.S. oil industry history". Possible senses for the term "history" in the YAGO ontology are:

1) \{history\} - the discipline that records and interprets past events involving human beings; "he teaches Medieval history"; "history takes the long view".
2) \{history (album), history\} – History Compilation album by Dune Released 2000 Recorded. …
3) \{history (song), history, history (michael jackson song)\} - Song by Michael Jackson from the album History. Released 1995. "History" is a song on Michael Jackson's album, History. …

And some more senses.

The senses 2) \{history (album), history\} and 3) \{history (song), history, history (michael jackson song)\} obtained a similarity value above the similarity value of the sense 1) \{history\}, which should obtain the largest similarity.

Due to this fact, we have to drop the respective description from the local context of a Wikipedia concept.

The second approach considers the related WordNet concepts as the local context of a Wikipedia concept. In other words, we ignore related Wikipedia concepts and take WordNet concepts only. For this purpose, we take the union of the own words, and the words and descriptions of the immediate WordNet concepts as local context.

To discover whether more or less depth into the WordNet level provides better results, we choose the following variations:

- We take only immediate WordNet concepts ⇔ depth one (the blue concepts in Figure 6-2).
- We take the immediate WordNet concepts and their immediate hypernyms and hyponyms in WordNet ⇔ depth two (the blue and the red concepts in Figure 6-2).

Every increasing of the search depth into the WordNet level could denote exponential increase of time in worst case. Tests with depth three showed that it is not applicable, due to the large running time; therefore we only consider depth one and depth two.
However, the depth into the WordNet level is not the only object that could affect the results. In the old word sense disambiguation, in case of all cosine similarity values (i.e., the similarity value of the local context of a candidate concept and the local context of the query term) becoming zero, the candidate concept containing the most frequently occurring word will be chosen. An improvement for this allocation could be a summation of the word frequencies of the own words for each candidate concept and choosing the concept with the highest sum instead of the concept with the most frequently occurring word.

By regarding these considerations we get the following four test situations:

1) Depth one into the WordNet level with frequency sum of the own words.
2) Depth one into the WordNet level with the most frequently occurring word out of the own words.
3) Depth two into the WordNet level with frequency sum of the own words.
4) Depth two into the WordNet level with the most frequently occurring word out of the own words.

These four test situations were launched respectively on 100 queries out of the TREC Terabyte 2004 + 2005 (see also Appendix A.1).
Table 6-1

<table>
<thead>
<tr>
<th>WordNet ontology</th>
<th>YAGO ontology</th>
<th>Depth one</th>
<th>Depth two</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>with sum</td>
<td>without sum</td>
</tr>
<tr>
<td>correct senses</td>
<td>34.6535</td>
<td>24.1546</td>
<td>26.5701</td>
</tr>
<tr>
<td>incorrect senses</td>
<td>32.1782</td>
<td>47.8261</td>
<td>45.4106</td>
</tr>
<tr>
<td>single senses</td>
<td>33.1683</td>
<td>28.0193</td>
<td>28.0193</td>
</tr>
</tbody>
</table>

Table 6-1 lists the number of correctly disambiguated senses, incorrectly disambiguated senses, and single senses. To have a better estimation of the quality of the results, the results of the old word sense disambiguation with the WordNet ontology are displayed on the left hand accordingly. Single senses are the unique available senses.4

Moreover, out of the used 100 queries 202 subsequences (i.e., n-grams out of the queries) could be matched to possible senses (i.e., concepts containing these subsequences) in the old ontology; 207 subsequences could be matched to possible senses in the new YAGO ontology.

The best result for the number of correct disambiguated senses with around 27 % was achieved with the second test configuration out of the former listing (i.e., depth one into the WordNet level with the most frequently occurring word out of the own words). Unfortunately this result is around 8 % worse than the result of the old approach.

Furthermore we have to note the unique senses. In the old approach they constitute around 33 % and in the new approach they constitute around 28 %. A deeper look into the correctness of these unique senses showed that around 95 % of them were actually correct senses for the regarded terms. That means to achieve the complete number of correct disambiguated senses we can add the amount of unique senses to the amount of correct senses and gain around 68 % for the old approach and around 55 % for the new configuration. Hence, the result for the new configuration is around 13 % worse than the result of the old approach with the WordNet ontology.

To find out why this result is worse, we observed all subsequences with their respective possible senses. Thus it appeared that Wikipedia concepts and WordNet concepts do not represent the same kind of senses generally; because Wikipedia concepts are rather individuals than word senses. For example, we consider the word "schizophrenia" with its possible senses:

---

4 If only one sense exist, this one will be chosen.
schizophrenic disorder, dementia praecox, schizophrenia, schizophrenic psychosis; any of several psychotic disorders characterized by distortions of reality and disturbances of thought and language and withdrawal from social contact.

schizophrenia, schizophrenia (sepultura album), schizophrenia (album); Schizophrenia is the second studio album by Brazilian thrash metal band Sepultura, released in 1987 through Cogumelo records.

The first one is a WordNet concept, which is a concrete word sense. The second one is a Wikipedia concept, which is an album named "schizophrenia"; this is an individual but not an actual word sense.

The problem of this fact for TopX is the possibility of deciding whether a subsequence is rather a word sense than an individual, or vice versa.

Moreover, a good effect which may not be disregarded in the tests for the new YAGO ontology was the number of phrase subsequences (i.e., subsequences with more than one word); a higher number of such subsequences could be matched to possible senses, where some of such subsequences could not be matched to possible senses in the WordNet ontology. See following examples:

• "gastric bypass": In the YAGO ontology there exists a concept that contains this subsequence; In WordNet only the two single terms "gastric" and "bypass" matched to possible senses.
• "international criminal court": It exists as concept in the YAGO ontology; whereas in WordNet only the two single terms "international" and "criminal court" yield to concepts.

In the old ontology only 21 phrase subsequences yield to candidate concepts, where 17 thereof are correct senses; in the YAGO ontology 42 phrase subsequences match to possible senses and 37 thereof are correct senses.

That means the word sense disambiguation in the new YAGO ontology achieves doubled number of matched phrase subsequences and the correctness of the respective disambiguated concepts rises up from around 81 % to around 88 %.

Let us return to the decision whether a subsequence is a word sense or an individual. The problem, which often occurred in the former tests, was that each set of candidate concepts contains much more Wikipedia concepts than WordNet concepts. This resulted in rather disambiguating Wikipedia concepts than WordNet concepts, although an existing WordNet concept actually would be the correct one.

Hence we introduced the following decision criterion:
If a WordNet concept belongs to the candidate concepts of a subsequence, this subsequence rather will be a word sense; otherwise, this subsequence rather will be an individual.

That way, the word sense disambiguation ignores the according Wikipedia candidate concepts if at least one WordNet candidate concept exists. This procedure was also preferred by YAGO during the information extraction (i.e., if a word was found in the WordNet level and in the Wikipedia level at the same time, the concept in the WordNet level was taken), also explained in [5].

However, to discover whether this decision criterion is applicable for TopX, we take a deeper look at the information extraction for the WordNet synsets for the YAGO ontology [5]. Special handling was done for the nouns of WordNet which appear as names of Wikipedia pages (i.e., a noun out of WordNet is also the name of a Wikipedia page, e.g., "Albert Einstein"). Since the Wikipedia page of such nouns is simply about the noun and does not describe an individual, which has the same name like the noun (e.g., "History" is a noun for WordNet, but an album title for Wikipedia), they always give preference to WordNet concepts and discard the Wikipedia individuals in such cases. There were around 15,000 cases, where a noun also appeared as a name of a Wikipedia page. Since YAGO decides that way, this decision criterion should be also applicable for the YAGO ontology in TopX. Critical cases for this decision would be senses which are only available within the Wikipedia level and could be ignored that way; however these cases are insignificantly small.

To discover whether this decision criterion enhances the result of the word sense disambiguation, we run a test with this decision criterion and the test configuration with best result in the former tests (i.e., depth one into the WordNet level with the most frequently occurring word out of the own words).

<table>
<thead>
<tr>
<th></th>
<th>WordNet ontology</th>
<th>YAGO ontology</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>with sum</td>
<td>without sum</td>
</tr>
<tr>
<td>correct senses</td>
<td>34.6535</td>
<td>40.0966</td>
<td>38.1643</td>
<td></td>
</tr>
<tr>
<td>incorrect senses</td>
<td>32.1782</td>
<td>31.8841</td>
<td>33.8164</td>
<td></td>
</tr>
<tr>
<td>single senses</td>
<td>33.1683</td>
<td>28.0193</td>
<td>28.0193</td>
<td></td>
</tr>
</tbody>
</table>

As statistically significant in Table 6-2 the amount of correct senses rises up from around 27 % (see Table 6-1) to around 40 %, which is also more than the amount of correct sense for the old approach which is around 35 %. Under-
standably, the amount of single senses stays constant with around 28 %. The number of correct senses and the number of single senses lead to around 68 % for the WordNet ontology and around 68 % for the YAGO ontology. That means with this decision criterion we could increase the result from 55 % up to 68 %, which is as good as the old approach.

The former tests showed that the word sense disambiguation is not an easy problem for the new ontology, since there exist several factors which can affect the results. However, in any case we could achieve the same precision as for the old approach and we enhanced the results for phrase subsequences around 7 %.
Chapter 7

Query Expansion for the YAGO Ontology

This section concerns with the adoptions, which have to be done for the query expansion on the new YAGO ontology.

As already mentioned, the table RELATIONS contains only edge similarity values (i.e., similarity values of two immediately related concepts). However, there remain the similarity values of paths (i.e., similarity values of concepts which are related through a sequence of edges); these will be computed during the query expansion.

Before we go on, we review the process of the query expansion (see also Section 2.3.2). It begins with a source concept - which is the result concept of the word sense disambiguation - and searches through all of its outgoing edges of a conditional type of relation (e.g., subclassof) to find related concepts and to add their words to the source query coevally. This process will be continued on each result concept of the word sense disambiguation, and on all found concepts if their similarity value is above a defined value $\delta$.

The similarity value $sim(S)$ of each found concept $S$ will be computed through the product of the similarity values $sim(e_j)$ of all visited edges $e_j$ on the respective path $\langle e_1, \ldots, e_n \rangle$ (i.e., the path from the source concept $S_i$ to the found concept $S$). As reminder, the similarity value between the source concept $S_i$ and the found concept $S$ is just the similarity value of the path $\langle e_1, \ldots, e_n \rangle$.

$$sim(S) = sim(\langle e_1, \ldots, e_n \rangle) = \prod_{j=1}^{n} sim(e_j)$$

Moreover, if two concepts are related over a single edge (i.e., the first found concepts beginning from the source concept), then their similarity value is equal to the similarity value of the appropriate edge.
As an example, we take a look at Figure 7-1. The similarity value of concept $H$ is defined by $\text{sim}(e) \cdot \text{sim}(f) \cdot \text{sim}(g)$ and the similarity value of concept $B$ is defined through $\text{sim}(a)$.

### 7.1 The Word Expansion

As already known, all words out of the concepts, whose similarity value is above $\delta$, will be added to the source query during the query expansion. At this point we have to handle the problem that already appeared during the similarity computation of edges, namely the occurrence of special signs (see also Section 4.5.3).

Hence, the first adaptation for the query expansion is the disregarding of words containing special signs.

### 7.2 NAGA Similarity

As seen before, the similarities of concepts, which are related through a sequence of edges, are computed during the query expansion through the product of the similarity values of the edges visited on the respective path.

Let us observe the similarity computation on concept $C$ in Figure 7-1. The similarity of the following concepts $F$ and $G$ is the product $\text{sim}(a) \cdot \text{sim}(b) \cdot \text{sim}(c)$ and $\text{sim}(a) \cdot \text{sim}(b) \cdot \text{sim}(d)$ respectively, where $\text{sim}(a) \cdot \text{sim}(b) = \text{sim}(C)$.

Hence, $\text{sim}(F) = \text{sim}(C) \cdot \text{sim}(c)$ and $\text{sim}(G) = \text{sim}(C) \cdot \text{sim}(d)$. In this manner, the similarity value $\text{sim}(C)$ has not to be recomputed by $\text{sim}(a) \cdot \text{sim}(b)$ for each of the following concepts $F$ and $G$; but only the value $\text{sim}(C)$ has to be available for $F$ and $G$. That means, the similarity values of the previous edges $a$
and $b$ are dispensable for the similarity computation of $F$ and $G$. In other words, each step of the query expansion needs only the similarity value $\text{sim}(N)$ of the respective previous visited node $N$ (i.e., $\text{sim}(D)$ in case of $E$, or $\text{sim}(C)$ in case of $F$ or $G$).

This procedure is fine as long as the similarity measure of paths is defined through a product; like it is the case for the Dice similarity measure and the Best WordNet measure (see also Section 4.1 and Section 4.3). However, the NAGA similarity measure (see also Section 4.2) is not a product, but the addition of the informativeness and the confidence of the respective path.

$$NAGA(S) = \beta \cdot \text{conf}(S) + (1 - \beta) \cdot \text{inf}(S)$$

where $\text{conf}(S)$ is the confidence and $\text{inf}(S)$ is the informativeness. That means the computation of the path similarity needs two values instead of one value. Therefore the similarity computation of paths has to be changed during the query expansion.

The informativeness and the confidence of a path are defined by the product of the respective values of the visited edges on this path (see Section 4.2). They are stored in the tables RELATIONS and RELATIONCONFIDENCES.

$$\text{inf}(S) = \prod_{j=1}^{n} \text{inf}(e_j)$$

and respectively,

$$\text{conf}(S) = \prod_{j=1}^{n} \text{conf}(e_j)$$
Figure 7-2 illustrates the similarity computation based on the NAGA measure. \(\text{sim}(B)\) is relevant for the decision, whether the words of \(B\) should be expanded to the query and whether the query expansion has to be continued for the subgraph spanned by \(B\). However, \(\text{sim}(B)\) is irrelevant for the computation of the similarity values of the concepts \(C\) and \(D\), since only the values \(\text{conf}(B)\) and \(\text{inf}(B)\) are relevant there.

Obviously, the formulas of \(\text{inf}(S)\) and \(\text{conf}(S)\) are very similar to the formula of the path similarity of the Dice similarity:

\[
\text{sim}(S) = \prod_{j=1}^{n} \text{sim}(e_j)
\]

That means, the path computation of the informativeness and the confidence act like the path computation of the Dice similarity, as already mentioned before.

Hence, we only have to adapt the path similarity computation during the query expansion as follows. Instead of one value (i.e., the similarity value of the concept) we have to hold two values, namely the informativeness and the confidence. For the decision whether the query expansion has to be continued for the subgraph spanned by a concept \(N\) and whether the words of \(N\) has to be added to the query, we need the final similarity value of \(N\); what is the addition of \(\text{inf}(S)\) and \(\text{conf}(S)\) of the previous visited concept \(S\) each weighted through \(\beta\), see also the formula above of \(\text{NAGA}(S)\).
It has to be noted, that this adaption is only valid for the NAGA measure; if the query expansion is based on another measure, the old method has to be applied.

7.3 Worked Samples

In this section we show some worked samples for the similarity computation during the query expansion.

**Dice Similarity Measure on the WordNet ontology**

\[
sim(e_1) = 0.0387 \quad \text{IS-A} \quad \sim \quad \text{Physicist} \quad \sim \quad \text{Scientist} \quad \sim \quad \text{Physicist} \quad \sim \quad \text{Scientists}\n\]

\[
sim(\text{Physicist}) = \sim(e_1) = 0.0387 \quad \sim \quad \text{sim(Scientist)} = \sim(\text{Physicist}) \sim \sim(e_2) = 0.0009
\]

**Figure 7-3** shows the path similarity on the WordNet ontology for the concepts *Albert Einstein* and *Scientist* based on the Dice similarity measure.

**Dice Similarity Measure on the YAGO ontology**

\[
sim(e_1) = 0.1124 \quad \text{IS-A} \quad \sim \quad \text{Physicist} \quad \sim \quad \text{Scientists}\n\]

\[
sim(\text{Physicist}) = \sim(e_1) = 0.1124 \quad \sim \quad \text{sim(Scientists)} = \sim(\text{Physicist}) \sim \sim(e_2) = 0.0171
\]

**Figure 7-4** shows some path similarities on the YAGO ontology for the concepts *Albert Einstein* and *Scientist*, *Albert Einstein* and *Hans Albert Einstein*, and *Albert Einstein* and *Germany* based on the Dice similarity measure.
**Chapter 7 Query Expansion for the YAGO Ontology**

**NAGA Similarity Measure on the YAGO ontology**

\[ \beta = 0.5 \]

\[ \text{sim}(\text{Scientist}) = 0.5 \cdot \text{conf}(e_1) \cdot \text{conf}(e_2) + 0.5 \cdot \text{inf}(e_1) \cdot \text{inf}(e_2) \]
\[ = 0.4971 \]

\[ \text{sim}(\text{Physicist}) = 0.5 \cdot \text{conf}(e_1) + 0.5 \cdot \text{inf}(e_1) \]
\[ = 0.6003 \]

\[ \text{sim}(\text{Germany}) = 0.5 \cdot \text{conf}(e_3) + 0.5 \cdot \text{inf}(e_1) \]
\[ = 0.7862 \]

Figure 7-5 shows some path similarities on the YAGO ontology for the concepts *Albert Einstein* and *Scientist*, *Albert Einstein* and *Hans Albert Einstein*, and *Albert Einstein* and *Germany* based on the NAGA similarity measure with influence of the confidence with \( \beta = 0.5 \).
Chapter 8

Conclusions

The YAGO ontology extends the knowledge base of TopX that way that more individuals and concepts exist. A further enhancement is the flexibility, due to the amount of different relation types in YAGO. Moreover, YAGO introduced other kinds of relation types which are, in comparison to the relation types of the WordNet ontology, not hierarchical (e.g., bornin or haswonprize).

However, the YAGO ontology makes some procedures like the word sense disambiguation and the query expansion a little bit catchier and brought some procedures, as for example the Dijkstra algorithm, to their limits due to the size of YAGO; hence these procedures had to be adapted or even reconstructed.

In Chapter 6 we showed that a lot of factors within YAGO could complicate and worsen the word sense disambiguation. But approaches like disregarding related Wikipedia concepts at the local context of a possible sense and the restriction on the disambiguation on senses in a level (i.e., in the WordNet level or in the Wikipedia level) made the precision of the disambiguation results as good as the precision for the old approach at the WordNet ontology; despite the fact that the difficulty increased. Furthermore, the amount of phrase subsequences out of the query terms, which could successfully be matched to possible senses in the YAGO ontology, rose up. Even the precision of the results of the word sense disambiguation in this context is around 7% better than the precision for the WordNet ontology.

In this thesis we introduced two further measures for the similarity on relations special for the YAGO ontology; these are the NAGA similarity measure and the Best WordNet measure. The evaluation of the computed similarity values for all measures (i.e., both new measures and the already used one for the WordNet ontology) in Chapter 5 denotes, that the NAGA measure and the Dice measure are applicable for the similarity computation of YAGO relations. However, the Best WordNet measure is most likely not applicable for the YAGO ontology due to the structure of the IS-A hierarchy of YAGO.

Altogether, we could adapt TopX to the restrictions of YAGO so that it is a further adaptive ontology. We even found a further possible measure for the relation similarities, the NAGA measure, which is significantly for the query
expansion and also crucial for a possible enhancement of the precision of resulting data items.

8.1 Open Issues & Future Work

8.1.1. Word Sense Disambiguation

In section Chapter 6 we listed some problems for the word sense disambiguation through the YAGO ontology. However, with some approaches we could eliminate these problems and could enhance the precision for the YAGO ontology up to 68 %, which is equal to the precision of the WordNet ontology.

But we think we could further enhance the precision of the word sense disambiguation on the YAGO ontology, especially on the Wikipedia level.

The disambiguation takes place in the Wikipedia level when there is no WordNet sense for a query term (see Section 6.2); however, we often have a larger amount of Wikipedia concepts in this case. In average, the amount is doubled with around 10 in comparison to the amount of terms in the WordNet level with around 5. There exist even some outliers with a very large number, as for example for the term "sue", where over 130 possible senses in the Wikipedia level exist.

It is clear that a disambiguation on a higher number of concepts is more difficult than a disambiguation on a smaller number of concepts and could even increase the probability of a worse disambiguation.

An enhancement in this context could be a grouping of the available concepts in the Wikipedia level. Conceivable groups could be "albums" or "persons", which actually could be obtained by the IS-A hierarchy of YAGO. If we consider a query term and could, due to the local context of the query term (see Section 2.3.1), decide to which group this query term belongs; we could restrict the word sense disambiguation not only to the level but also to the respective group.

As an example, we search for possible senses for the query term "sue", which has no possible senses in the WordNet level; however, there exist a huge amount of possible senses in the Wikipedia level. These possible senses are albums containing the term "sue" or persons named by "sue" and many more. The classification on the query term "sue" into one of the possible groups, as for example the persons group, would decrease the amount of possible senses that way that only senses being persons would be regarded. This decrease could denote an enhancement on the precision of the word sense disambiguation in the Wikipedia level.
8.1.2. Query Expansion

The configuration for the query expansion currently only includes the IS-A relations subclassof, type, and ispartof for the YAGO ontology (i.e., hypernym, hyponym and part_meronym for the WordNet ontology). However, a lot of other relation types exist in YAGO, which could be interesting for the query expansion.

Some of these could be the relation types connecting to locations. As an example, let us search for a person in Germany, while a person with the same name could also live in another country. In such cases, it could be helpful to expand for the birthplace of a person, which can be obtained by the YAGO relation bornin, to obtain those data items which contain information about the correct person. Further such interesting information could be values which can be obtained by the so-called value relations (see Section 2.2), as for example birthdays or area sizes.

Let us return to the relation types which connect to locations (e.g., bornin). A special relation type in this context is locatedin, which structures the ordering of the locations in YAGO. This knowledge could also be interesting for the query expansion. Assuming the relation types, which connect to locations, are considered for the query expansion. A possible expansion in this context could be ("Albert Einstein", bornin, "Ulm"), when searching for relevant pages of "Albert Einstein". However, it could be that most of the pages do not contain the word "Ulm", but the word "Germany". That means, it would be helpful to get the upper location (i.e., upper in the IS-A hierarchy) of a location (i.e., in case of "Ulm" it is "Germany"), which can be obtained by the relation type locatedin. Also the opposite direction, which denotes the lower locations (i.e., lower in the IS-A hierarchy) of a location (e.g., listing of some towns in "Germany") could be conceivable. However, we have to take into account that the respective relations for the opposite direction have to be weighted in their significance, so that only the n most significant terms will be extracted (e.g., only "Berlin" in case of "Germany"), due to the higher amount of destinations (since a country can have a huge amount of towns).

8.1.3. Value Relations

Actually, value concepts (e.g., area sizes) depend on defined measures (e.g., miles). However, what is if another measure of a value is required (e.g., meters instead of miles)? A further enhancement of TopX and the YAGO ontology could be a translation of some measures into SI units [35] (e.g., from miles to
Conclusions

meter) and vice versa by introducing further relation types or functions. Such a relation type or function could be named by "miles2meter" or "meter2miles".

Further useful information which is currently only implicit available in TopX, is the information of the used measure for a value.

If such information could be retrieved, also the query expansion could be enhanced in this context.

8.1.4. Relation Similarities

The three currently available similarity measures (see Chapter 4) use corpus information or ontology information. A further interesting point for the similarity computation could be semantic similarities.

The relation types, which connect to locations, seem to be good candidates for a semantic similarity. Therefore, the area size of locations or their population density could be obtained to compute a similarity value. An adequate knowledge base for these information could be World Gazetteer [34].

The logic for such a semantic similarity could be: "a location with smaller area size is more relevant than a location with a higher area size" in case of locatedin forward and vice versa for locatedin backward (e.g., for "Germany" and locatedin backward the largest town of Germany, "Berlin", should be chosen). However, has to be regarded that some of these relation types also connect places like museums. That means, for such kind of locations certainly exist no area size or population density.

8.1.5. Evaluation of the Query Expansion

The distribution of the similarity values as evaluated in Chapter 5 shows that the distributions for the NAGA values and the Dice values closely agree with the distribution of the similarity values for the WordNet ontology. That means it suggests itself that these measures could be good candidates for the query expansion on the YAGO ontology. However, we have to keep in mind that the YAGO ontology is much bigger than the WordNet ontology. In more details, the query expansion will expand, for same $\delta$, much more concepts for the YAGO ontology than for the WordNet ontology.

To get really evidence if these measures are applicable or not, there, some benchmark tests on the precision of the returned data have to be run.
In these tests, the following aspects should be considered:

- Relation types: including further relation types other than IS-A relations (e.g., value relations or location relations).
- Restrictions on the amount of included words of a concept: if a concept will be expanded, all words of it will be expanded to the original query terms (see Section 2.3.2). Since the concepts in the YAGO ontology also contains irrelevant words like typing errors, the precision for relevant pages could be worse due to such words.
- Manually disambiguated concepts: the word sense disambiguation is already good with a precision of 68%. However, to obtain the correct precision of the query expansion, the prior word sense disambiguation should be as good as possible, in order to avoid a worsening due to some incorrect disambiguations. This can be achieved by manually mapping the correct concepts for all query terms and for all subsequences of the query terms.
- Smoothing factor: actually there exists a smoothing function on the relation similarity values, which smoothes the respective values. This smoothing seems to enhance the results of the query expansion for the WordNet ontology. It has to be investigated, whether this smoothing also enhances the results for the YAGO ontology or if it does the opposite.
Appendix A

A.1. TREC Terabyte Queries

701  <title>U.S. oil industry history</title>  <desc>Describe the history of the U.S. oil industry</desc>
702  <title>Pearl farming</title>  <desc>Pearl farming operations: actual farming operations described, culturing pearls, "Japanese pearl productions," status of pearl farming, production.</desc>
703  <title>U.S. against International Criminal Court</title>  <desc>What are the arguments the U.S. uses against joining the International Criminal Court?</desc>
704  <title>Green party political views</title>  <desc>What are the goals and political views of the Green Party.</desc>
705  <title>Iraq foreign debt reduction</title>  <desc>Identify any efforts, proposed or undertaken, by world governments to seek reduction of Iraq's foreign debt.</desc>
706  <title>Controlling type II diabetes</title>  <desc>What are methods used to control type II diabetes?</desc>
707  <title>Aspirin cancer prevention</title>  <desc>What evidence is there that aspirin may help prevent cancer?</desc>
708  <title>Decorative slate sources</title>  <desc>What are sources of slate stone for decorative use?</desc>
709  <title>Horse racing jockey weight</title>  <desc>What are the limits and regulations concerning jockey weight in horse racing?</desc>
710  <title>Prostate cancer treatments</title>  <desc>What are the various treatments for prostate cancer?</desc>
711  <title>Train station security measures</title>  <desc>What security measures have been employed at train stations due to heightened security concerns?</desc>
712  <title>Pyramid scheme</title>  <desc>What are some actual examples of pyramid schemes?
<num> 713
<title>Chesapeake Bay Maryland clean</title>
<desc>What is the state of Maryland doing to clean up the Chesapeake Bay?</desc>
<num> 714
<title>Licence restrictions older drivers</title>
<desc>What restrictions are placed on older persons renewing their drivers' licenses in the U.S.?</desc>
<num> 715
<title>Schizophrenia drugs</title>
<desc>What organizations (private or governmental) are developing drugs to combat schizophrenia?</desc>
<num> 716
<title>Spammer arrest sue</title>
<desc>Have any spammers been arrested or sued for sending unsolicited e-mail?</desc>
<num> 717
<title>Gifted talented student programs</title>
<desc>What states or localities offer programs for gifted and talented students?</desc>
<num> 718
<title>Controlling acid rain</title>
<desc>What methods are used to control acid rain and its effects?</desc>
<num> 719
<title>Cruise ship damage sea life</title>
<desc>What kinds of harm do cruise ships do to sea life such as coral reefs, and what is the extent of the damage?</desc>
<num> 720
<title>Federal welfare reform</title>
<desc>Find documents about Federal welfare reform legislation, regulation, and policy.</desc>
<num> 721
<title>Census data applications</title>
<desc>What applications are there for U.S. decennial census data, and how is it used?</desc>
<num> 722
<title>Iran terrorism</title>
<desc>In what ways does Iran support terrorism?</desc>
<num> 723
<title>Executive privilege</title>
<desc>What is the U.S. government's definition of 'executive privilege'?</desc>
<num> 724
<title>Iran Contra</title>
<desc>What was the Iran-Contra scandal and what were the consequences?</desc>
<num> 725
<title>Low white blood cell count</title>
<desc>What would cause a lowered white blood cell count?</desc>
<num> 726
<title>Hubble telescope repairs</title>
<desc>What repairs have been made on the Hubble telescope?
A.1 TREC Terabyte Queries

<table>
<thead>
<tr>
<th>num</th>
<th>title</th>
<th>desc</th>
</tr>
</thead>
<tbody>
<tr>
<td>727</td>
<td>Church arson</td>
<td>Identify any specific instances of church arson.</td>
</tr>
<tr>
<td>728</td>
<td>Whales save endangered</td>
<td>What's being done to save endangered whales?</td>
</tr>
<tr>
<td>729</td>
<td>Whistle blower department of defense</td>
<td>What have been revelations of whistle blowers concerning the U.S. Department of Defense?</td>
</tr>
<tr>
<td>730</td>
<td>Gastric bypass complications</td>
<td>What are some of the possible complications and potential dangers of gastric bypass surgery?</td>
</tr>
<tr>
<td>731</td>
<td>Kurds history</td>
<td>What is the history of the Kurds?</td>
</tr>
<tr>
<td>732</td>
<td>U.S. cheese production</td>
<td>What cheese production is carried out in the U.S.?</td>
</tr>
<tr>
<td>733</td>
<td>Airline overbooking</td>
<td>What are the regulations regarding airline overbooking?</td>
</tr>
<tr>
<td>734</td>
<td>Recycling successes</td>
<td>What recycling projects have been successful?</td>
</tr>
<tr>
<td>735</td>
<td>Afghan women condition</td>
<td>Is the condition of Afghan women better under the new government than under the Taliban?</td>
</tr>
<tr>
<td>736</td>
<td>location BSE infections</td>
<td>Where have animals infected with bovine spongiform encephalopathy (also known as BSE or Mad Cow disease) been found?</td>
</tr>
<tr>
<td>737</td>
<td>Enron California energy crisis</td>
<td>What allegations have been made about Enron's culpability in the California Energy crisis?</td>
</tr>
<tr>
<td>738</td>
<td>Anthrax hoaxes</td>
<td>What are some examples of anthrax hoaxes?</td>
</tr>
<tr>
<td>739</td>
<td>Habitat for Humanity</td>
<td>What is the organization &quot;Habitat for Humanity&quot;, and what activities are they involved in?</td>
</tr>
<tr>
<td>740</td>
<td>regulate assisted living Maryland</td>
<td>Who regulates assisted living facilities in Maryland?</td>
</tr>
</tbody>
</table>
Appendix A

<num> 741
<title> Artificial Intelligence
<desc> What is artificial intelligence?
<num> 742
<title> hedge funds fraud protection
<desc> What protection do investors have against fraud by hedge funds?
<num> 743
<title> Freighter ship registration
<desc> What are the regulations and other considerations concerning registering a freighter in a country?
<num> 744
<title> Counterfeit ID punishments
<desc> What punishments or sentences have been given in the U.S. for making or selling counterfeit IDs?
<num> 745
<title> Doomsday cults
<desc> Identify any doomsday cult, their name, and location throughout the world.
<num> 746
<title> Outsource job India
<desc> What jobs have been outsourced to India?
<num> 747
<title> Library computer oversight
<desc> What control or oversight is there over computer use in public libraries?
<num> 748
<title> Nuclear reactor types
<desc> Name the types of nuclear reactor power plants in operation in the United States.
<num> 749
<title> Puerto Rico state
<desc> Do people in Puerto Rico want for it to become a U.S. State?
<num> 750
<title> John Edwards womens issues
<desc> What are Senator John Edwards' positions on women's issues such as pay equity, abortion, Title IX and violence against women.
<num> 751
<title> Scrabble Players
<desc> Give information on Scrabble players, when and where Scrabble is played, and how popular it has been.
<num> 752
<title> Dam removal
<desc> Where have dams been removed and what has been the environmental impact?
<num> 753
<title> bullying prevention programs
<desc> What programs have been used in schools to prevent bullying of students.
<table>
<thead>
<tr>
<th>num</th>
<th>title</th>
<th>desc</th>
</tr>
</thead>
<tbody>
<tr>
<td>754</td>
<td>domestic adoption laws</td>
<td>Provide any legal information about domestic human adoption.</td>
</tr>
<tr>
<td>755</td>
<td>Scottish Highland Games</td>
<td>What is the history and location of Scottish highland games in the United States.</td>
</tr>
<tr>
<td>756</td>
<td>Volcanic Activity</td>
<td>Locations of volcanic activity which occurred within the present day boundaries of the U.S. and its territories.</td>
</tr>
<tr>
<td>757</td>
<td>Murals</td>
<td>Show examples of murals.</td>
</tr>
<tr>
<td>758</td>
<td>Embryonic stem cells</td>
<td>What are embryonic stem cells, and what restrictions are placed on their use in research?</td>
</tr>
<tr>
<td>759</td>
<td>civil war battle reenactments</td>
<td>When and where are Civil War battle reenactments held?</td>
</tr>
<tr>
<td>760</td>
<td>American Muslim mosques schools</td>
<td>Statistics regarding American Muslims, mosques, and schools.</td>
</tr>
<tr>
<td>761</td>
<td>Problems of Hmong Immigrants</td>
<td>Describe the problems faced by Hmong immigrants to the United States</td>
</tr>
<tr>
<td>762</td>
<td>History of Physicians in America</td>
<td>Who have been considered &quot;doctors&quot; since the first European settlement in America?</td>
</tr>
<tr>
<td>763</td>
<td>Hunting deaths</td>
<td>Give information on human deaths associated with hunting for game.</td>
</tr>
<tr>
<td>764</td>
<td>Increase mass transit use</td>
<td>What is being done to increase mass transit use?</td>
</tr>
<tr>
<td>765</td>
<td>Ephedra ma huang deaths</td>
<td>How many deaths have been attributed to the drug ephedra, also known as the herbal ingredient ma huang?</td>
</tr>
<tr>
<td>766</td>
<td>Diamond smuggling</td>
<td>Illicit activity involving diamonds, to include diamond smuggling.</td>
</tr>
<tr>
<td>767</td>
<td>Pharmacist License requirements</td>
<td></td>
</tr>
</tbody>
</table>
Appendix A

<desc> What are the requirements for a pharmacist's license in the U.S.?
<num> 768
<title> Women in state legislatures
<desc> What is the number of women legislators or what percentage of the total legislators in any given state are women?
<num> 769
<title> Kroll Associates Employees
<desc> Identify employees of Kroll Associates.
<num> 770
<title> Kyrgyzstan-United States relations
<desc> What is the state of Kyrgyzstan-United States relations?
<num> 771
<title> deformed leopard frogs
<desc> What deformities have been found in leopard frogs?
<num> 772
<title> flag display rules
<desc> What are the rules or guidelines for display of the United States flag?
<num> 773
<title> Pennsylvania slot machine gambling
<desc> What is the legal status of slot machine gambling in Pennsylvania?
<num> 774
<title> Causes of Homelessness
<desc> What are some of the causes of homelessness?
<num> 775
<title> Commercial candy makers
<desc> Identify commercial candy makers and give information concerning them.
<num> 776
<title> Magnet schools success
<desc> Are magnet schools considered successful in districts where they have been created?
<num> 777
<title> hybrid alternative fuel cars
<desc> What hybrid or alternative fuel passenger cars are auto manufacturers now marketing or developing for future sales?
<num> 778
<title> Golden ratio
<desc> Golden ratio formula, description, or examples.
<num> 779
<title> Javelinas range and description
<desc> Describe the Javelina or collared peccary and its geographic range.
<num> 780
<title> Arable land
<desc> How much of planet Earth is arable at present? Area must have plenty of water, sun and soil to support plant life.
<num> 781
<titl> Squirrel control and protections
<desc> Give information on steps to manage, control, or protect squirrels.
<num> 782
<titl> Orange varieties seasons
<desc> What are the varieties of oranges and when is each in season?
<num> 783
<titl> school mercury poisoning
<desc> How have mercury poisonings of children occurred in schools and what measures are being taken to prevent such incidents?
<num> 784
<titl> mersenne primes
<desc> Give a definition or description of Mersenne prime numbers.
<num> 785
<titl> Ivory-billed woodpecker
<desc> What is the history and present status of the Ivory-billed Woodpecker?
<num> 786
<titl> Yew trees
<desc> Where do yew trees grow anywhere on the globe?
<num> 787
<titl> Sunflower Cultivation
<desc> Give information on the cultivation of sunflowers.
<num> 788
<titl> Reverse mortgages
<desc> What are reverse mortgages and how do they work?
<num> 789
<titl> abandoned mine reclamation
<desc> Find information on abandoned mine reclamation projects.
<num> 790
<titl> women's rights in Saudi Arabia
<desc> Provide any description of laws or restrictions affecting Saudi Arabian women's rights.
<num> 791
<titl> Gullah geechee language culture
<desc> Describe the historical background and present status of Gullah-Geechee language and culture
<num> 792
<titl> Social Security means test
<desc> Does Social Security use a means test?
<num> 793
<titl> Bagpipe Bands
<desc> Give information on, and examples of, bagpipe bands.
<num> 794
<titl> pet therapy
<desc> How are pets or animals used in therapy for humans and what are the benefits?
Appendix A

<num> 795
<title>notable cocker spaniels</title><desc>Provide any reference to notable cockers or other spaniels.</num> 796
<title>Blue Grass Music Festival history</title><desc>Describe the history of bluegrass music and give location of bluegrass festivals.</num> 797
<title>reintroduction of gray wolves</title><desc>Where in the US have gray wolves been reintroduced in the wild?</num> 798
<title>Massachusetts textile mills</title><desc>History, development, and locations of textile mills in Massachusetts</num> 799
<title>Animals in Alzheimer's research</title><desc>What animals have been used in Alzheimer's research?</num> 800
<title>Ovarian Cancer Treatment</title><desc>The remedies and treatments given to lesson or stop effects of ovarian cancer.
Bibliography


Bibliography


[26] Sebastian Michel. Top-k Aggregation Queries in Large-Scale Distributed Systems.


[34] http://world-gazetteer.com/